



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: Análisis del *software* ImageJ para el análisis científico de imágenes

AUTOR: Daniel Sánchez Valenciano

TITULACIÓN: Grado en Ingeniería de Sonido e Imagen

TUTOR: Alfonso Luís Martín Marcos.

DEPARTAMENTO: Ingeniería Audiovisual y Comunicaciones

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: Carlos Carrillo Sánchez

VOCAL: Alfonso Luís Martín Marcos

SECRETARIO: José Manuel Díaz López

Fecha de lectura:

Calificación:

El Secretario,

Análisis del *software* ImageJ para el análisis científico de imágenes

Daniel Sánchez Valenciano

*Estudiante de Grado en Ingeniería de Sonido e Imagen
Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación
Universidad Politécnica de Madrid*

Resumen

ImageJ es un programa informático de tratamiento digital de imagen orientado principalmente hacia el ámbito de las ciencias de la salud. Se trata de un *software* de dominio público y de código abierto desarrollado en lenguaje Java en las instituciones del *National Institutes of Health* de Estados Unidos. Incluye por defecto potentes herramientas para editar, procesar y analizar imágenes de casi cualquier tipo y formato. Sin embargo, su mayor virtud reside en su extensibilidad: las funcionalidades de ImageJ pueden ampliarse hasta resolver casi cualquier problema de tratamiento digital de imagen mediante *macros*, *scripts* y, especialmente, *plugins* programables en lenguaje Java gracias a la API que ofrece. Además, ImageJ cuenta con repositorios oficiales en los que es posible obtener de forma gratuita *macros*, *scripts* y *plugins* aplicables en multitud de entornos gracias a la labor de la extensa comunidad de desarrolladores de ImageJ, que los depura, mejora y amplía frecuentemente.

Este documento es la memoria de un proyecto que consiste en el análisis detallado de las herramientas de tratamiento digital de imagen que ofrece ImageJ. Tiene por objetivo determinar si ImageJ, a pesar de estar más enfocado a las ciencias de la salud, puede resultar útil en el entorno de la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación de la Universidad Politécnica de Madrid, y en tal caso, resaltar las características que pudieran resultar más beneficiosas en este ámbito y servir además como guía introductoria. En las siguientes páginas se examinan una a una las herramientas de ImageJ (versión 1.48q), su funcionamiento y los mecanismos subyacentes. Se sigue el orden marcado por los menús de la interfaz de usuario: el primer capítulo abarca las herramientas destinadas a la manipulación de imágenes en general (menú *Image*); el segundo, las herramientas de procesado (menú *Process*); el tercero, las herramientas de análisis (menú *Analyze*); y el cuarto y último, las herramientas relacionadas con la extensibilidad de ImageJ (menú *Plugins*).

Abstract

ImageJ is a digital image processing computer program which is mainly focused at the health sciences field. It is a public domain, open source software developed in Java language at the National Institutes of Health of the United States of America. It includes powerful built-in tools to edit, process and analyze almost every type of image in nearly every format. However, its main virtue is its extensibility: ImageJ functionalities can be widened to solve nearly every situation found in digital image processing through macros, scripts and, specially, plugins programmed in Java language thanks to the ImageJ API. In addition, ImageJ has official

repositories where it is possible to freely get many different macros, scripts and plugins thanks to the work carried out by the ImageJ developers community, which continuously debug, improve and widen them.

This document is a report which explains a detailed analysis of all the digital image processing tools offered by ImageJ. Its final goal is to determine if ImageJ can be useful to the environment of Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación of Universidad Politécnica de Madrid, in spite of being focused at the health sciences field. In such a case, it also aims to highlight the characteristics which could be more beneficial in this field, and serve as an introductory guide too. In the following pages, all of the ImageJ tools (version 1.48q) are examined one by one, as well as their work and the underlying mechanics. The document follows the order established by the menus in ImageJ: the first chapter covers all the tools destined to manipulate images in general (menu Image); the second one covers all the processing tools (menu Process); the third one includes analyzing tools (menu Analyze); and finally, the fourth one contains all those tools related to ImageJ extensibility (menu Plugins).

Tabla de contenidos

1.	Introducción	7
2.	Menú Image.....	10
2.1.	Submenú Type	10
2.2.	Submenú Adjust.....	32
2.3.	Show Info.....	49
2.4.	Properties.....	50
2.5.	Submenú Color	51
2.6.	Submenú Stacks.....	59
2.7.	Submenú Hyperstacks.....	78
2.8.	Crop.....	80
2.9.	Duplicate.....	80
2.10.	Rename	81
2.11.	Scale.....	81
2.12.	Submenú Transform.....	81
2.13.	Submenú Zoom.....	85
2.14.	Submenú Overlay.....	86
2.15.	Submenú Lookup Tables.....	88
3.	Menú Process.....	90
3.1.	Smooth	90
3.2.	Sharpen	92
3.3.	Find Edges	93
3.4.	Find Maxima.....	94
3.5.	Enhance Contrast	96
3.6.	Submenú Noise	97

3.7.	<i>Submenú Shadows</i>	100
3.8.	<i>Submenú Binary</i>	101
3.9.	<i>Submenú Math</i>	107
3.10.	<i>Submenú FFT</i>	111
3.11.	<i>Submenú Filters</i>	118
3.12.	<i>Submenú Batch</i>	120
3.13.	<i>Image Calculator</i>	120
3.14.	<i>Subtract Background</i>	121
3.15.	<i>Repeat Command</i>	123
4.	Menú Analyze	124
4.1.	<i>Measure</i>	124
4.2.	<i>Analyze Particles</i>	132
4.3.	<i>Summarize</i>	134
4.4.	<i>Distribution</i>	135
4.5.	<i>Label</i>	136
4.6.	<i>Clear Results</i>	136
4.7.	<i>Set Measurements</i>	136
4.8.	<i>Set Scale</i>	137
4.9.	<i>Calibrate</i>	139
4.10.	<i>Histogram</i>	140
4.11.	<i>Plot Profile</i>	143
4.12.	<i>Surface Plot</i>	144
4.13.	<i>Submenú Gels</i>	146
4.14.	<i>Submenú Tools</i>	150
5.	Plugins	168
5.1.	<i>Submenú Macros</i>	168

5.2.	<i>Submenú Shortcuts</i>	173
5.3.	<i>Submenú Utilities</i>	174
5.4.	<i>Submenú New</i>	178
5.5.	<i>Compile and Run</i>	193
5.6.	<i>Install</i>	193
6.	Conclusiones	194
7.	Referencias	195

1. INTRODUCCIÓN

ImageJ es un programa informático de tratamiento digital de la imagen en general, aunque está más enfocado al ámbito de las ciencias de la salud, en donde suele encontrar aplicación. Es un *software* de dominio público y de código abierto desarrollado en lenguaje Java en las instituciones del grupo *National Institutes of Health* de Estados Unidos. Su autor principal, Wayne Rasband, trabaja como voluntario especial en el *Research Services Branch* del *National Institute of Mental Health*. Se puede ejecutar en forma de *applet*, como aplicación ejecutable y en general en cualquier plataforma que disponga de una máquina virtual Java (versión 5 o superior). Se distribuye de forma gratuita para sistemas operativos de las familias Windows, MAC OS y Linux desde el apartado correspondiente de su sitio web, <http://imagej.nih.gov/ij/download.html>.

ImageJ proporciona una gran cantidad de herramientas de edición, procesado y análisis de imagen aplicables a imágenes de tipo *8-bit*, *16-bit* y *32-bit*, y de multitud de formatos (jpg, bmp, png, gif, tiff, dicom, fits, raw,...). En cuanto a la edición, ImageJ permite convertir imágenes de un tipo a otro, hacer ajustes avanzados sobre el brillo y el contraste de las imágenes, umbralizarlas, modificar sus dimensiones, inspeccionar minuciosamente sus propiedades, gestionar las características específicas de las imágenes en color y de las imágenes indexadas, administrar las imágenes mediante pilas e hiperpilas, llevar a cabo sobre ellas todo tipo de transformaciones geométricas, crear y gestionar recubrimientos sobre las imágenes,...

En lo que se refiere al procesado, ImageJ provee de multitud de filtros espaciales y del dominio de la frecuencia: filtros suavizantes, filtros agudizadores, filtros de detección de bordes, filtros espaciales cuyo *kernel* es directamente determinable por el usuario, filtros tridimensionales, filtros paso banda, filtros del dominio de la frecuencia que también son totalmente configurables por el usuario,...Es capaz de hacer convoluciones y transformadas de Fourier. Proporciona utilidades para eliminar (y añadir) varios tipos de ruido. Contiene un menú específico para llevar a cabo operaciones matemáticas y lógicas con los píxeles de las imágenes, otro en el que se agrupan las herramientas destinadas al procesado de imágenes binarias y otro más para las operaciones que involucran más de una imagen de forma simultánea. Además, hay que mencionar que ImageJ soporta procesamiento por lotes y que se trata de un programa multiproceso.

Con respecto al análisis, hay que destacar que ImageJ es capaz de llevar a cabo un gran número de medidas sobre las imágenes o sobre zonas específicas de éstas: histogramas, perfiles, medidas de área, niveles de brillo medios, desviaciones estándar, máximos, mínimos, modas, centroides, centros de masas, perímetros, diámetros de Feret, asimetrías, curtosis, distancias, ángulos,... Tiene herramientas para hacer análisis automáticos de objetos en imágenes binarias, herramientas para calibrar las imágenes (tanto espacialmente como en densidad) y ofrece un conjunto de utilidades específicas para el análisis de geles electroforéticos.

Sin embargo, las funcionalidades que hacen de ImageJ un programa realmente potente y que lo han llevado a ser popular son aquellas relacionadas con su extensibilidad. Todas las herramientas descritas hasta este punto conforman el núcleo de ImageJ, pero éste puede crecer de forma prácticamente ilimitada gracias a *macros*, *scripts* y, especialmente, *plugins*. ImageJ tiene un lenguaje *macro* propio, es capaz de interpretar el lenguaje JavaScript y ofrece una API (*Application Programming Interface* o Interfaz de Programación de Aplicaciones) para programar *plugins* en lenguaje Java que se integran con facilidad en ImageJ y que aprovechan sus funcionalidades. Además, existe un amplísimo repertorio gratuito *online* de *macros*, *scripts* y *plugins* que crece a diario gracias a la comunidad de desarrolladores de ImageJ. De hecho, existen distribuciones de ImageJ en las que se incluyen por defecto distintas selecciones de *plugins* en función del objetivo (FIJI es el mejor ejemplo). Todo esto hace que, a pesar de estar orientado a las ciencias de la salud, ImageJ puede extenderse hasta resultar aplicable en casi cualquier ámbito y pueda resolver casi cualquier problema de tratamiento digital de imagen.

Para acceder a todas sus funcionalidades, ImageJ despliega la interfaz de usuario que se muestra en la Figura 2.1.1:

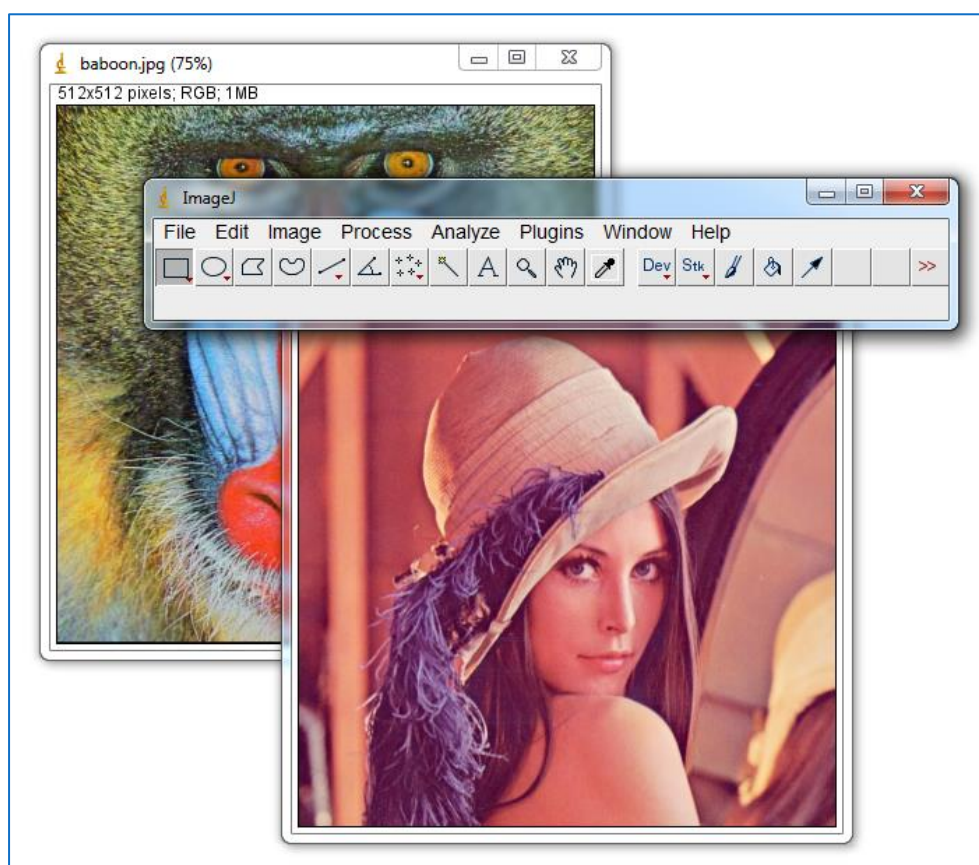


Figura 2.1.1. Interfaz de usuario de ImageJ junto a varias ventanas con imágenes

Como se puede observar en la Figura 2.1.1, ImageJ consta de una sencilla interfaz de usuario con una barra de menús y una barra de herramientas (además de una barra de estado que aparece vacía en la figura). En la barra de menús se pueden encontrar los menús que se encontrarían en cualquier aplicación (Archivo, Edición, Ventana y Ayuda) y los menús con las herramientas de tratamiento digital de imagen ordenadas según se ha comentado en los párrafos anteriores. La barra de herramientas está destinada, principalmente, a albergar las herramientas de selección: selección rectangular, selección elíptica, selección poligonal, selección libre, selección lineal,...

El presente proyecto pretende analizar detalladamente ImageJ a partir de los conocimientos y destrezas adquiridos en la rama de imagen del Grado en Ingeniería de Sonido e Imagen. Tiene por fin último determinar la conveniencia de utilizar ImageJ, que como ya se ha comentado suele encontrar aplicación en las ciencias de la salud, como herramienta de tratamiento digital de imagen en general en la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación de la Universidad Politécnica de Madrid, y en tal caso, señalar qué ventajas podría presentar con respecto a otras alternativas, destacar las funcionalidades que pudieran resultar más beneficiosas en este ámbito y, por último, servir como guía introductoria.

En las siguientes páginas, que sirven de memoria del análisis llevado a cabo, se examinan las herramientas de tratamiento digital de imagen que ofrece ImageJ (concretamente, la versión 1.48q) siguiendo el orden que marcan los menús de su interfaz de usuario. En primer lugar, se examinan las herramientas relacionadas con la edición y la manipulación de imágenes en general incluidas en el menú *Image*; en segundo lugar, se estudian todas las herramientas de procesado agrupadas en el menú *Process*; a continuación, se tratan las utilidades relacionadas con el análisis estadístico recogidas en el menú *Analyze*, y para terminar, se analizan las funcionalidades relacionadas con la extensibilidad incluidas en el menú *Plugins*; el resto de menús no se analizan por no estar directamente relacionados con el tratamiento digital de imagen, sino con la gestión del programa.

2. MENÚ IMAGE

El menú *Image* aglutina una gran cantidad de herramientas relacionadas con la imagen en general: herramientas para el estudio y el ajuste de las propiedades de las imágenes, herramientas para administrar pilas e hiperpilas, herramientas para llevar a cabo operaciones geométricas con las imágenes, herramientas para gestionar los recubrimientos de las imágenes, etc. A continuación se estudia detalladamente cada una de ellas en el orden en que se muestran en el menú *Image* y en sus submenús.

2.1. Submenú Type

El submenú *Type* (Tipo) permite realizar las siguientes operaciones:

- Consultar el tipo de imagen al que pertenece la imagen activa. Para ello basta con desplegar el submenú y comprobar qué tipo de imagen de entre los que se listan aparece marcado con un *tic* negro (ver Figura 2.1.1):

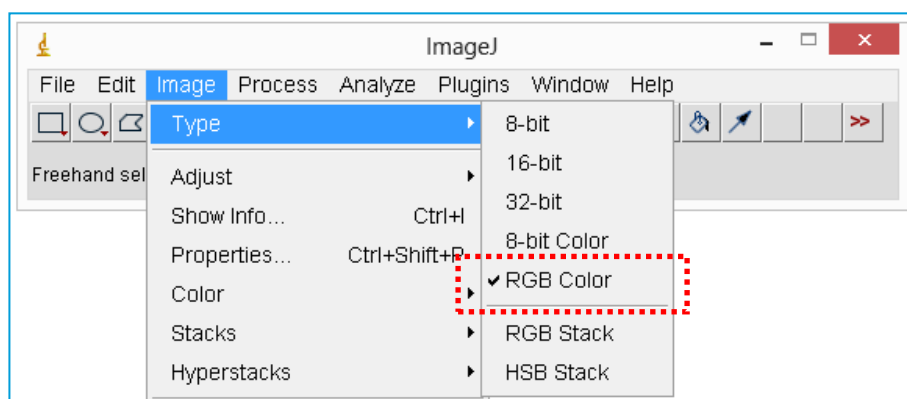


Figura 2.1.1. El submenú *Type* indica que, en este caso, la imagen activa es de tipo *RGB Color*

- Convertir la imagen activa a otro tipo de imagen. Para ello sólo es necesario desplegar el submenú y seleccionar el tipo de imagen al que se desea convertir la imagen activa. En caso de que la conversión no sea posible se muestra un cuadro de diálogo informativo que lista las conversiones viables. Las distintas opciones, su compatibilidad y su funcionamiento se detallan a continuación.

2.1.1. 8-bit

La opción *8-bit* convierte la imagen activa a una imagen en escala de grises con 8 bits de profundidad, lo que significa que la nueva imagen puede tener hasta $2^8 = 256$ niveles de gris, desde el nivel 0 (negro) hasta el nivel 255 (blanco). Los tipos de imagen desde los que se puede realizar una conversión a *8-bit* son los siguientes:

- Imagen original de tipo *16-bit*. Es decir, una imagen en escala de grises con hasta $2^{16} = 65536$ niveles de gris. Para realizar una conversión desde *16-bit* hasta *8-bit* se aplica a todos los píxeles de la imagen de tipo *16-bit* una correspondencia lineal entre su margen dinámico real (desde su nivel mínimo hasta su nivel máximo) y el margen dinámico máximo de una imagen de tipo *8-bit* (desde el nivel 0 hasta el nivel 255, para aprovechar todo el margen dinámico). De este modo, la relación entre los niveles de brillo de los píxeles de las imágenes originales y convertidas es la siguiente:

$$v = \frac{255}{\max - \min}(u - \min) + 0 \quad (2.1.1)$$

donde u es el nivel de brillo de los píxeles de la imagen a convertir, \min y \max son, respectivamente, el mínimo y el máximo nivel de brillo de los píxeles de la imagen a convertir, y v es el nivel de brillo de los píxeles de la imagen convertida. Dado que los niveles de brillo de una imagen de tipo *8-bit* son enteros, los valores de v deben redondearse al entero más próximo (en realidad, como se comprueba a continuación, se redondea al entero inmediatamente superior).

A modo de ejemplo, en la Figura 2.1.2 se muestra una imagen en formato *.tiff* y de tipo *16-bit*, con nivel de brillo mínimo de 0 y un nivel de brillo máximo de 10110. En el momento de la captura, el puntero estaba situado sobre el píxel de coordenadas $x=149$ e $y=256$, que tenía un nivel de brillo de 2431.

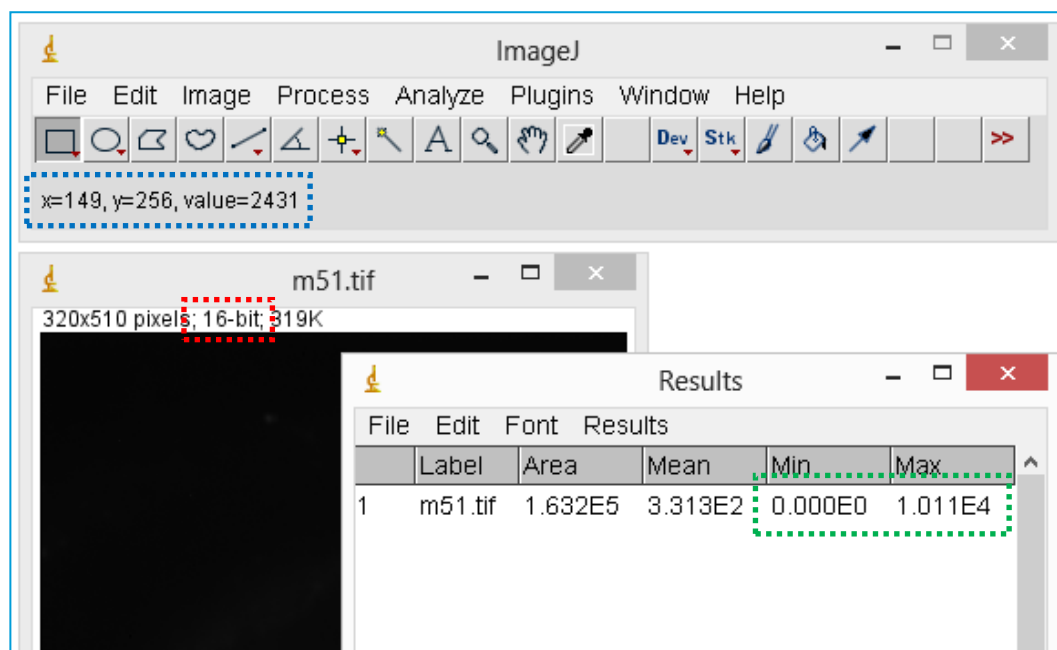


Figura 2.1.2. Imagen de tipo *16-bit* (destacado en rojo). En la tabla de resultados se muestra que su nivel de brillo mínimo es 0 y su nivel de brillo máximo es 10110 (destacado en verde). En la barra de estado de ImageJ se aprecia que el puntero está situado sobre el píxel de coordenadas $x=149$ e $y=256$, y que éste tiene un nivel de brillo de 2431 (destacado en azul)

Según la expresión (2.1.1), si la imagen de la Figura 2.1.2 se convirtiera a una imagen de tipo *8-bit*, el nivel de brillo del píxel de coordenadas $x=149$ e $y=256$ debería ser:

$$v = \frac{255}{10110 - 0}(2431 - 10110) + 255 = 61,3 \approx 62 \quad (2.1.2)$$

En la Figura 2.1.3 se aprecia la imagen una vez convertida a tipo *8-bit*, en la que el nuevo nivel mínimo es 0 y el nuevo nivel máximo es 255. Además, como se esperaba, el nivel de brillo del píxel de coordenadas $x=149$ e $y=256$ en la imagen convertida a tipo *8-bit* es 62.

También se puede apreciar en la Figura 2.1.3 que el tamaño de la nueva imagen, a pesar de tener el mismo número de píxeles (320x510), es aproximadamente la mitad: 159 KB frente a los 319 KB de la imagen original. Evidentemente, esto se debe a que los datos que corresponden exclusivamente a imagen, que son la mayoría, ocupan la mitad, puesto que si antes el nivel de brillo de cada píxel se describía con 16 bits, ahora se describe sólo con la mitad: 8 bits. Esta ventajosa disminución de tamaño lleva aparejada, por el contrario, una desventaja: en la nueva imagen la información se describe de forma menos precisa, sólo con 256 niveles de gris, cuando antes se disponía de 65536 niveles. Esta nueva precisión, si bien es suficiente para aquellas imágenes destinadas a ser procesadas por el sistema visual humano, puede resultar escasa en el ámbito del procesamiento científico de imágenes, en el que diferencias de brillo reducidas pueden llegar a ser muy significativas (como se aprecia en el estudio de cierto tipo de imágenes tratadas en capítulos posteriores).

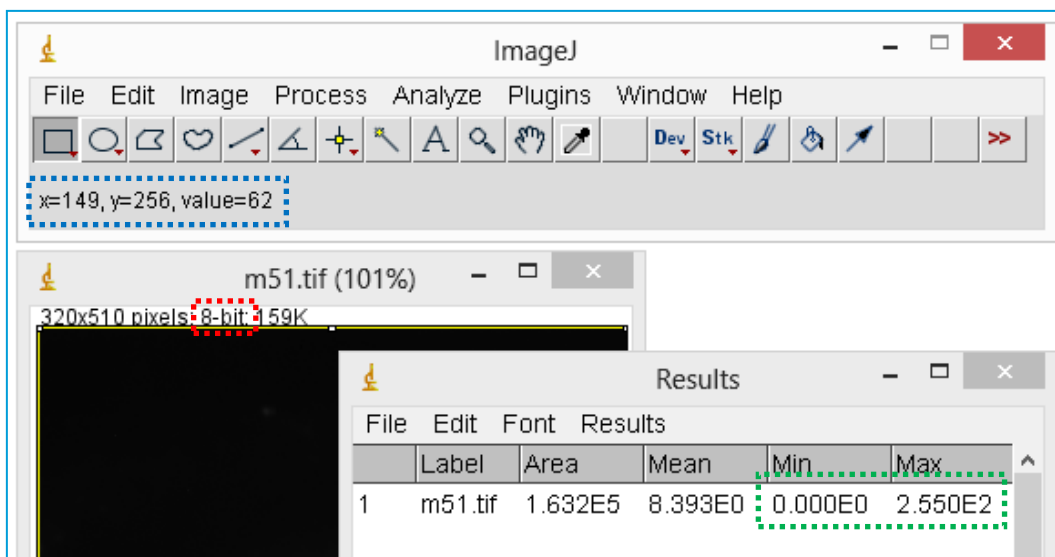


Figura 2.1.3. Imagen de tipo *8-bit* (destacado en rojo). En la tabla de resultados se muestra que su nivel de brillo mínimo es 0 y su nivel de brillo máximo es 255 (destacado en verde). En la barra de estado de ImageJ se aprecia que el puntero está situado sobre el píxel de coordenadas $x=149$ e $y=256$, y que éste tiene un nivel de brillo de 62 (destacado en azul)

Es necesario mencionar que el escalamiento del margen dinámico sólo se lleva a cabo si la opción *Scale When Converting* (Escalar Al Convertir) del apartado *Conversions* del submenú *Options*, menú *Edit*, está activa, lo cual ocurre por defecto. Sin embargo, en caso contrario se omite el escalamiento, y en la nueva imagen se mantienen los valores de los niveles de gris de los píxeles, salvo aquellos que superan 255, que pasan a valer 255, y aquellos que están por debajo de 0 (lo cual puede ocurrir en imágenes de tipo *16-bit signed*), que pasan a valer 0. En la Figura 2.1.4 se muestra el resultado de convertir la imagen de la Figura 2.1.2 sin escalar el margen dinámico. Esta desadaptación implica que la nueva imagen aparece saturada; como en la imagen original existía una gran cantidad de píxeles con niveles de brillo superiores a 255, en la nueva imagen aparecen muchas zonas “quemadas”, en las que todos los píxeles tienen un nivel de gris igual, 255, a pesar de que en la imagen original no tenían el máximo nivel de gris.

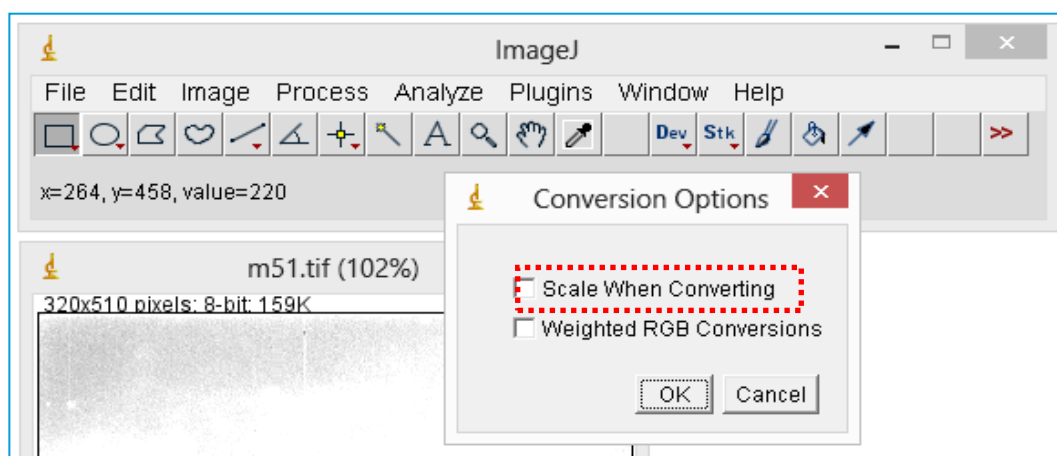


Figura 2.1.4. Imagen de tipo *16-bit* convertida a *8-bit* sin haber escalado el margen dinámico

Por último, hay que señalar que la conversión de imágenes de tipo *16-bit* a imágenes de tipo *8-bit* no sólo soporta imágenes, sino también pilas de imágenes (los conceptos de pila e hiperpila de imágenes se estudian de forma más detallada en los apartados correspondientes).

- Imagen original de tipo *32 bit*. En este caso el tipo de imagen origen es tal que puede llegar a contener $2^{32} = 4.294.967.296$ niveles de gris. Hay que señalar que en las imágenes de tipo *32-bit* los niveles se representan en punto flotante, por lo que los niveles son números reales y no sólo enteros. En cualquier caso, la conversión de imágenes de tipo *32-bit* a *8-bit* es completamente análoga a la conversión de imágenes de tipo *16-bit* a *8-bit* estudiada anteriormente en todos los aspectos (también soporta la conversión de pilas de imágenes).
- Imagen original de tipo *8-bit Color*. En este tipo de imagen a cada píxel también le corresponden 8 bits, pero a diferencia de las imágenes de tipo *8-bit*, en las imágenes de tipo *8-bit Color* los 8 bits no describen un nivel de gris, sino un índice de una tabla LUT (*Lookup Table* o Tabla de Consulta). En esta tabla hay por tanto $2^8 = 256$ entradas, y cada una contiene la descripción de un color en base a los niveles de brillo

de sus componentes RGB (*Red, Green and Blue*, o Rojo, Verde y Azul), que a su vez se describen cada una con 8 bits (es decir, en una LUT se pueden describir 256 colores a elegir entre $2^8 \cdot 2^8 \cdot 2^8 = 16.777.216$). Así que, para cada píxel, los 8 bits indican una entrada de la tabla LUT en la que se describe el color que le corresponde.

A pesar de que esta opción está disponible, realmente no lleva a cabo una conversión de *8-bit Color* a *8-bit*; una vez procesada, la imagen sigue siendo *8-bit Color*, pero su LUT se sustituye por otra en la que hay 256 niveles de grises ordenados desde el negro al blanco: en la entrada con índice 0 se describe el negro, con R=0, G=0 y B=0; en la entrada con índice 1 se describe un gris con R=1, G=1 y B=1; y así sucesivamente hasta el blanco (R=255, G=255 y B=255) en el final de la tabla. Esto se ilustra en la Figura 2.1.5: a pesar de estar en escala de grises, sigue siendo una imagen de tipo *8-bit Color*, puesto que sigue teniendo una LUT y los 8 bits asociados a cada píxel indican índices de esta tabla y no niveles de gris. La diferencia es que la LUT cambia en el segundo caso, y sólo contiene colores grises, que además crecen de forma lineal con el índice.

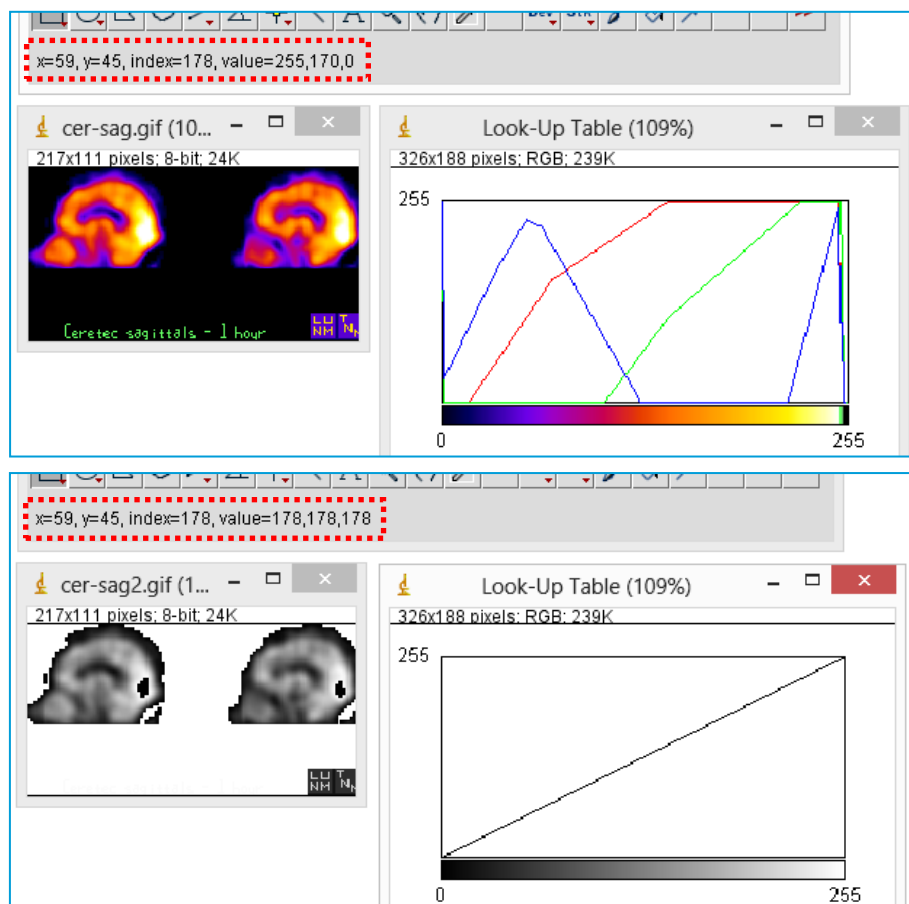


Figura 2.1.5. Arriba: imagen de tipo *8-bit Color*. Abajo: imagen anterior “convertida” en imagen de tipo *8-bit*

Como se puede apreciar en la Figura 2.1.5, los índices no cambian con la conversión: el píxel de coordenadas x=59 e y=45 siempre tiene índice 178, pero en la LUT de la imagen original este índice se

asocia a un color con $R=255$, $G=170$ y $B=0$, y por el contrario en la LUT de la imagen procesada el índice 178 corresponde al gris de nivel 178 ($R=G=B=178$). Para realizar una verdadera conversión de *8-bit Color* a *8-bit* es necesario convertir primero la imagen de tipo *8-bit Color* a una imagen de tipo *RGB Color*, para después convertir ésta a *8-bit*. Estas conversiones se discuten a continuación.

Para terminar hay que añadir que también soporta la conversión de pilas de imágenes.

- Imagen original de tipo *RGB Color*. Este tipo de imagen tiene color. Como ya se ha avanzado, en ella el color de cada píxel se describe en función de los niveles de sus tres componentes primarias RGB que, mediante mezcla aditiva, permiten formarlo. A cada componente le corresponden 8 bits, de modo que cada uno puede tener 2^8 niveles distintos, y por lo tanto en una imagen de este tipo podrían llegar a encontrarse $2^8 \cdot 2^8 \cdot 2^8 = 16.777.216$ colores diferentes, correspondientes a todas las combinaciones posibles de valores de las componentes RGB.

Una imagen de tipo *RGB Color* contiene de forma intrínseca información de color (también llamada crominancia) y de brillo (también llamada luminancia, Y), y es éste último el que se tiene en cuenta para realizar la conversión a una imagen en escala de grises, como es el escaso de las imágenes de tipo *8-bit*. La información del brillo o luminancia se puede extraer llevando a cabo una media ponderada entre las tres componentes RGB. En este caso, la conversión se lleva a cabo siguiendo las reglas de construcción de las señales YCbCr descritas en la recomendación BT.601-6¹ de la UIT-R (Unión Internacional de Telecomunicación, sector de Radiocomunicaciones). En concreto, para la construcción de la luminancia Y se recomienda atenerse a la siguiente expresión:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (2.1.3)$$

Para hacer la conversión, en ImageJ se utiliza la siguiente aproximación (y, evidentemente, se redondea el resultado):

$$Y = 0.30 \cdot R + 0.59 \cdot G + 0.11 \cdot B \quad (2.1.4)$$

A modo de ejemplo, en la Figura 2.1.6 se muestra una conversión de una imagen de tipo *RGB Color* a *8-bit*. Además, se recoge la información de un píxel antes y después de la conversión. Cuando aún se trataba de una imagen en color, a éste se le asociaban tres valores, los de sus componentes RGB: 196, 84

¹ Parámetros de codificación de televisión digital para estudios con formatos de imagen normal 4:3 y de pantalla ancha 16:9

y 82 respectivamente. Al pasar a ser una imagen de tipo *8-bit*, sólo se le asocia el valor de su nivel de brillo: 117. Esto concuerda con la ecuación (2.1.4):

$$Y = 0.30 \cdot 196 + 0.59 \cdot 84 + 0.11 \cdot 82 = 117,38 \approx 117 \quad (2.1.5)$$

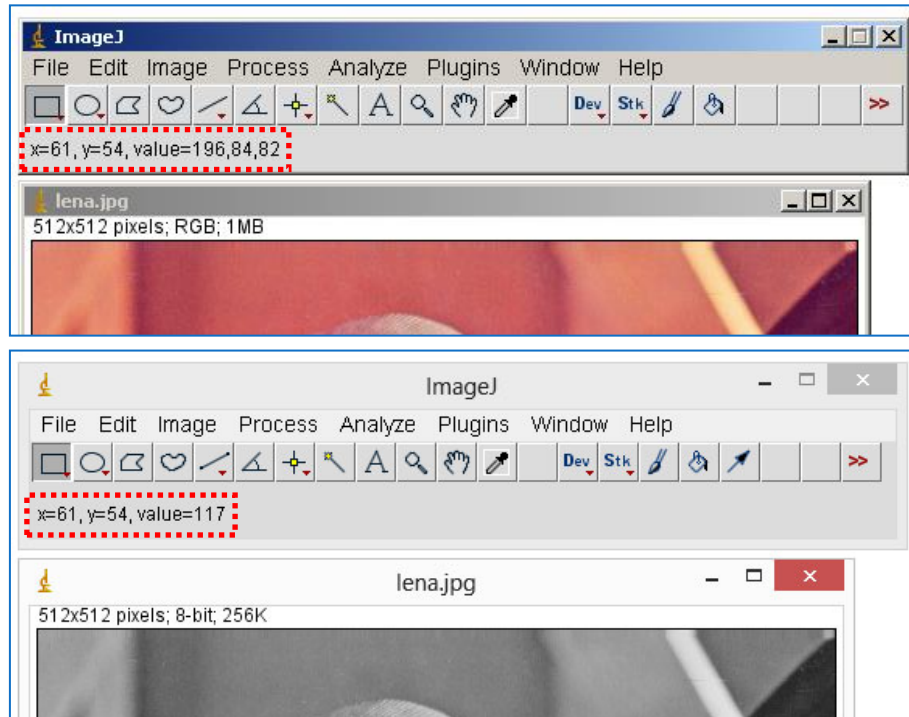


Figura 2.1.6. Arriba: imagen de tipo *8-bit Color*. Abajo: imagen anterior convertida en una imagen de tipo *8-bit*. En la barra de estado se aprecia que el puntero está situado sobre el píxel de coordenadas $x=59$ e $y=45$, al cual corresponden en el primer caso tres componentes RGB de valores $R=196$, $G=84$ y $B=82$, y en el segundo caso un único valor, el de su nivel de gris: 117 (destacado todo en rojo).

Sin embargo, también existe la posibilidad de que la conversión se lleve a cabo sin seguir la recomendación UIT-R BT.601-6. Para ello, debe estar desmarcada la casilla “*Weighted RGB conversions*” en el apartado *Conversions* del submenú *Options*, en el menú *Edit*. En este caso, la conversión se lleva a cabo haciendo una media sin ponderar de las componentes RGB, tal y como se aprecia en Figura 2.1.7, en la que se observa que, haciendo la conversión sin utilizar la media ponderada, el nuevo nivel de gris del píxel de coordenadas $x=61$ e $y=54$ (que en la imagen original tenía coordenadas $R=196$, $G=84$, $B=82$) es 121:

$$Y = \frac{196 + 84 + 82}{3} \approx 120,6667 \approx 121 \quad (2.1.6)$$

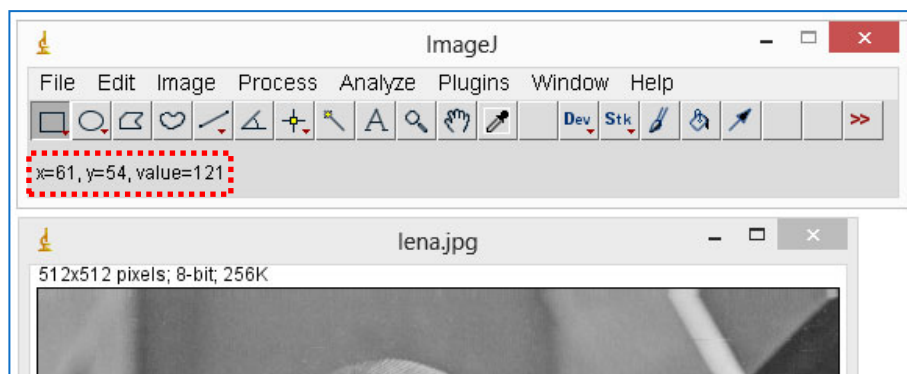


Figura 2.1.7. Imagen de la Figura 2.1.6 convertida a tipo *8-bit* sin usar una media ponderada. Se destaca en rojo el nuevo valor del mismo píxel destacado también en dicha figura.

Igual que en los casos anteriores, esta opción también sirve para convertir pilas de imágenes.

2.1.2. 16-bit

Esta opción permite convertir la imagen activa a una imagen en escala de grises con 16 bits de profundidad, es decir, con hasta $2^{16} = 65536$ niveles de gris. Es posible realizar conversiones a tipo *16-bit* desde los tipos *8-bit*, *32-bit* y *RGB Color*, y no sólo de imágenes sencillas, sino también de pilas de imágenes (salvo en el caso de origen *RGB Color*):

- Imagen original de tipo *8-bit*. A pesar de que es posible convertir una imagen de tipo *8-bit* en una imagen de tipo *16-bit* no se realiza ningún tipo de escalado entre los márgenes dinámicos y simplemente se trasladan los niveles de brillo de los píxeles de la imagen original a los nuevos píxeles. Para ejemplificarlo, se transforma la imagen que se muestra la Figura 2.1.7, que es de tipo *8-bit*, a tipo *16-bit*, y se muestra en la Figura 2.1.8. En ella se aprecia que, a pesar de que el tipo de imagen cambia a *16-bit* y podrían representarse más niveles de gris, se conservan los niveles de la imagen original (en la barra de estado se muestra el valor del mismo píxel que se inspeccionaba en la Figura 2.1.7).

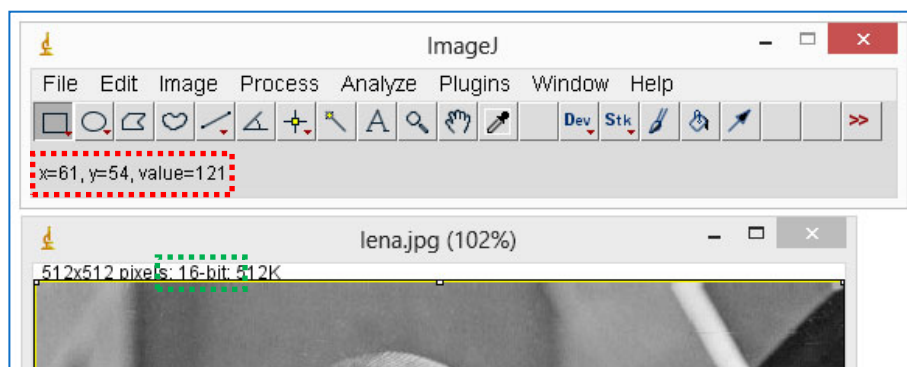


Figura 2.1.8. Imagen de tipo *8-bit* convertida a tipo *16-bit*, lo cual se destaca en verde. En rojo se destaca el valor del píxel de coordenadas $x=61$, $y=54$, que se mantiene con respecto a la imagen original

- Imagen original de tipo *32-bit*. En esta conversión sí se lleva a cabo un escalamiento entre los márgenes dinámicos de las imágenes (siempre que esté marcada la opción *Scale When Converting*, como ya se ha

comentado). En la Figura 2.1.9 se muestra el código correspondiente a esta conversión. En él se aprecia que, para cada píxel de la imagen original de tipo *32-bit*, se lleva a cabo la siguiente operación:

$$v = (u - \min) \cdot \text{scale} = \frac{65535}{(\max - \min)}(u - \min) \quad (2.1.7)$$

```

if ((max-min)==0.0)
    scale = 1.0;
else
    scale = 65535.0/(max-min);
double value;
for (int i=0,j=0; i<width*height; i++) {
    if (doScaling)
        value = (pixels32[i]-min)*scale;

```

Figura 2.1.9. Código correspondiente a la conversión *32-bit* a *16-bit*

- Imagen original de tipo *RGB Color*. Esta conversión equivale a convertir primero la imagen de tipo *RGB Color* a una imagen de tipo *8-bit* para después convertir ésta en una imagen de tipo *16-bit*. El resultado es, por tanto, el mismo que se obtiene con la conversión de *8-bit* a *16-bit*: se trasladan los niveles de gris directamente y sin escalar a los píxeles de la nueva imagen.

2.1.3. 32-bit

Con esta opción se consigue convertir la imagen activa a una imagen en escala de grises de 32 bits de punto flotante y con signo, así que la imagen se podría describir con niveles reales positivos y negativos. Es posible realizar la conversión desde los siguientes tipos de imágenes:

- Imagen original de tipo *8-bit*. Al igual que sucedía en el caso de la conversión de *8-bit* a *16-bit*, no se lleva a cabo ningún tipo de escalamiento, simplemente se trasladan los niveles de gris de los píxeles de la imagen original a los nuevos píxeles. Esto se muestra en la Figura 2.1.10, que recoge la imagen utilizada en las figuras anteriores convertida a tipo *32-bit*, y que sigue conservando el valor de sus píxeles: el píxel de coordenadas $x=61$ e $y=54$ sigue teniendo un nivel de gris de 121,0. Sin embargo, se puede apreciar que efectivamente se trata de una imagen de tipo *32-bit* de punto flotante en el hecho de que aparecen decimales en los valores: aunque el valor del píxel en cuestión sigue siendo 121, el nivel ya no es un número entero, sino que tiene un decimal, aunque vale 0.

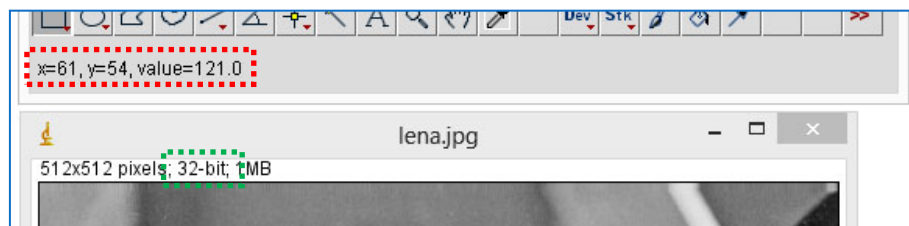


Figura 2.1.10. Imagen de tipo *8-bit* convertida a tipo *32-bit*, lo cual se destaca en verde. En rojo se destaca el valor del píxel de coordenadas $x=61$, $y=54$, que se mantiene con respecto a la imagen original

- Imagen original de tipo *16-bit*. Esta conversión es completamente análoga al caso anterior de *8-bit* a *32-bit*.
- Imagen original de tipo *RGB Color*. Al igual que sucedía con la conversión *RGB Color* a *16-bit*, esta conversión es equivalente a convertir primero la imagen de tipo *RGB Color* a una imagen de tipo *8-bit*, para después convertir ésta a una imagen de tipo *32-bit*. Sin embargo hay una diferencia, y es que, al ser el tipo de destino *32-bit* de punto flotante y con signo, se conservan los decimales en vez de redondear. Esto se demuestra en la Figura 2.1.11. En ella se muestra el resultado de convertir la imagen original *RGB Color* (ver Figura 2.1.6) directamente a *32-bit*, y se detalla una vez más la información correspondiente al píxel de coordenadas $x=61$ e $y=54$: en este caso no tiene el nivel de gris 121, sino el nivel 120,6667, que es el resultado sin redondear de hacer la media de los valores de las componentes RGB del píxel en cuestión en la imagen en color original, como se demuestra en la ecuación (2.1.6).

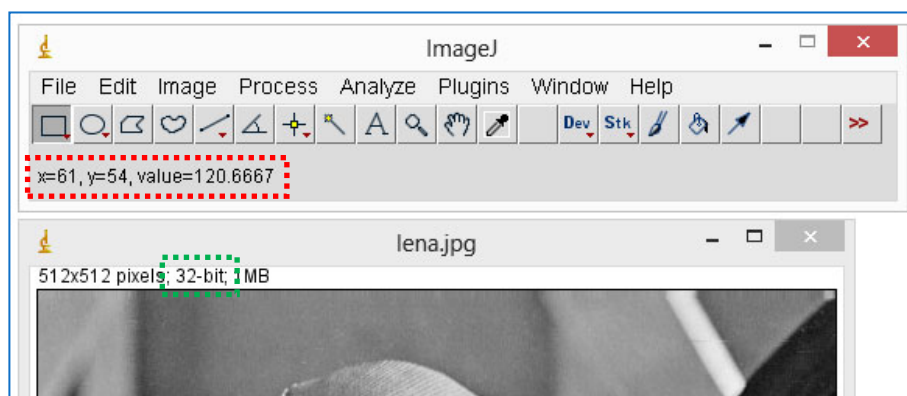


Figura 2.1.11. Imagen de tipo *RGB Color* convertida a tipo *32-bit*, lo cual se destaca en verde. En rojo se destaca el valor del píxel de coordenadas $x=61$, $y=54$

Salvo en el caso de las imágenes de tipo *RGB Color*, es posible aplicar las conversiones anteriores a pilas de imágenes.

2.1.4. 8-bit Color

Como ya se ha comentado, las imágenes de tipo *8-bit Color* son imágenes en color, pero los bits que corresponden a sus píxeles no describen los niveles de sus componentes RGB, sino los índices de una tabla LUT en la que se albergan los colores que pueden tomar los píxeles de la imagen.

Lógicamente, sólo es posible hacer verdaderas conversiones a tipo *8-bit Color* desde imágenes en color, es decir, desde imágenes de tipo *RGB Color* (ya sean imágenes sencillas o pilas de imágenes). Para llevar a cabo esta conversión es necesario reducir el número de colores de la imagen original, que podría ser de hasta $2^{24} = 16.777.216$, hasta como mucho los $2^8 = 256$ que caben en la tabla LUT, e intentando siempre conservar la información cromática que contiene la imagen con tanta precisión como sea posible. ImageJ emplea para ello el algoritmo *Heckberts median-cut color quantization algorithm*.

Este algoritmo pretende que cada uno de los colores que se recojan en la nueva LUT represente aproximadamente al mismo número de colores de la imagen original. Para ello trabaja sobre el concepto de espacio colorimétrico: un espacio en el que las dimensiones se corresponden con las componentes de los colores (en este caso, con las componentes RGB) y en el que todos los colores que puede haber en una imagen *RGB Color* están encerrados en un cubo de lado 255 (el máximo nivel que pueden tomar las componentes), como se muestra en la Figura 2.1.12. En este escenario, cada uno de todos estos colores se puede representar por un punto concreto del espacio RGB de coordenadas R_0, G_0, B_0 .

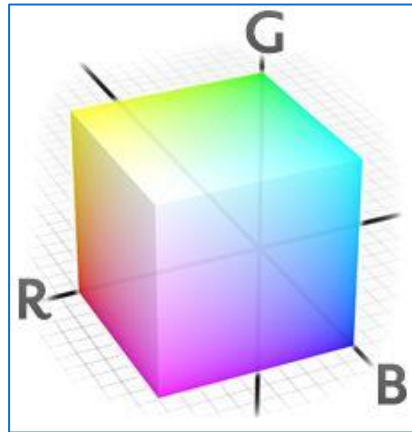


Figura 2.1.12. Espacio colorimétrico RGB

Grosso modo, el algoritmo consiste, en primer lugar, en encontrar el paralelepípedo más pequeño que encierra a todos los colores de la imagen en cuestión (es decir, que encierra los puntos del espacio colorimétrico que representan a todos los colores de la imagen original). A continuación, éste se divide en dos nuevos paralelepípedos de la siguiente forma: los colores encerrados, representados por puntos de tres coordenadas, se ordenan a lo largo de la mayor dimensión, y el paralelepípedo se divide en otros dos paralelepípedos con un plano perpendicular a dicha dimensión que la corta por la mediana de los puntos ordenados (de ahí el nombre de *median-cut*), de forma que a cada lado del plano hay aproximadamente la misma cantidad de puntos. Esta operación se ilustra con el siguiente ejemplo: se dispone de una imagen con 11 colores, cuyas coordenadas en el espacio colorimétrico se recogen en la Tabla 2.1.1, y sobre la que se va a aplicar la operación.

Tabla 2.1.1. Componentes RGB de los 11 colores de la imagen ficticia usada para ilustrar el algoritmo *Heckberts median-cut color quantization algorithm*

#	R	G	B
1	239	80	44
2	197	60	144
3	202	40	162
4	243	134	237

#	R	G	B
5	7	129	117
6	7	147	197
7	109	185	59
8	69	237	88
9	56	209	228
10	253	108	76
11	48	203	197

En el espacio de color, los puntos que representan los colores de la Tabla 2.1.1 quedarían encerrados en el paralelepípedo que se representa en la Figura 2.1.13:

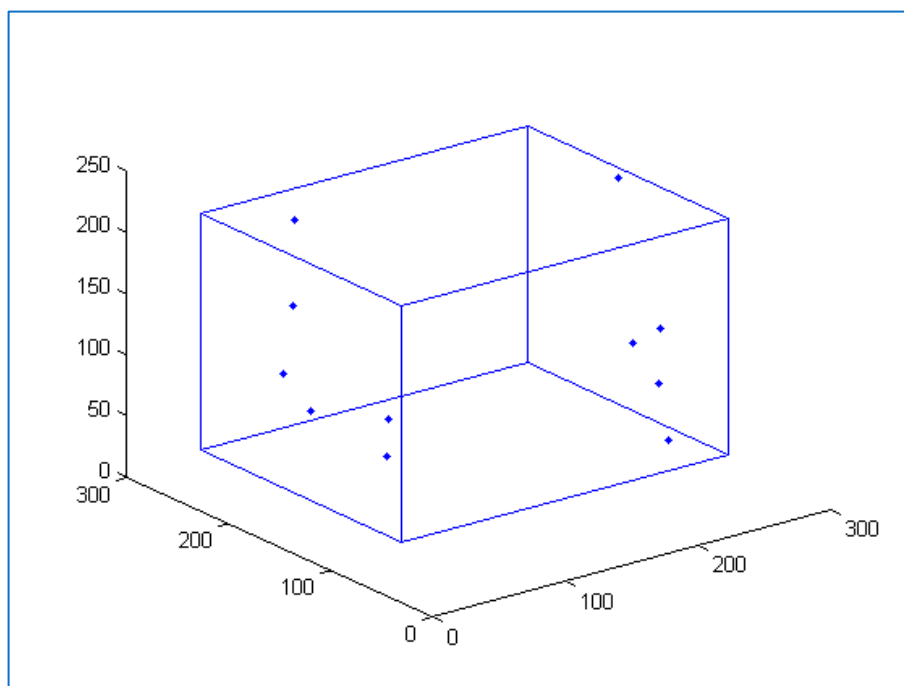


Figura 2.1.13. Colores de la Tabla 2.1.1 encerrados en el paralelepípedo en el espacio de color

Este paralelepípedo tiene dimensiones $R=246$, $G=197$ y $B=193$. Es decir, la mayor dimensión del paralelepípedo es la R , y por tanto es la que se usa para ordenar los puntos representativos de los colores. En la Tabla 2.1.2 aparecen ordenados de esta forma:

Tabla 2.1.2. Colores de la Tabla 2.1.1 ordenados según la dimensión R

#	R	G	B
5	7	129	117
6	7	147	197

#	R	G	B
11	48	203	197
9	56	209	228
8	69	237	88
7	109	185	59
2	197	60	144
3	202	40	162
1	239	80	44
4	243	134	237
10	253	108	76

El siguiente paso, como se ha comentado, sería dividir el paralelepípedo en otros dos con un plano perpendicular a la dimensión R que la corte por la mediana de los puntos ordenados, que se destaca sombreada en azul en la Tabla 2.1.2: 109. El resultado de dividir el paralelepípedo se muestra en la Figura 2.1.14:

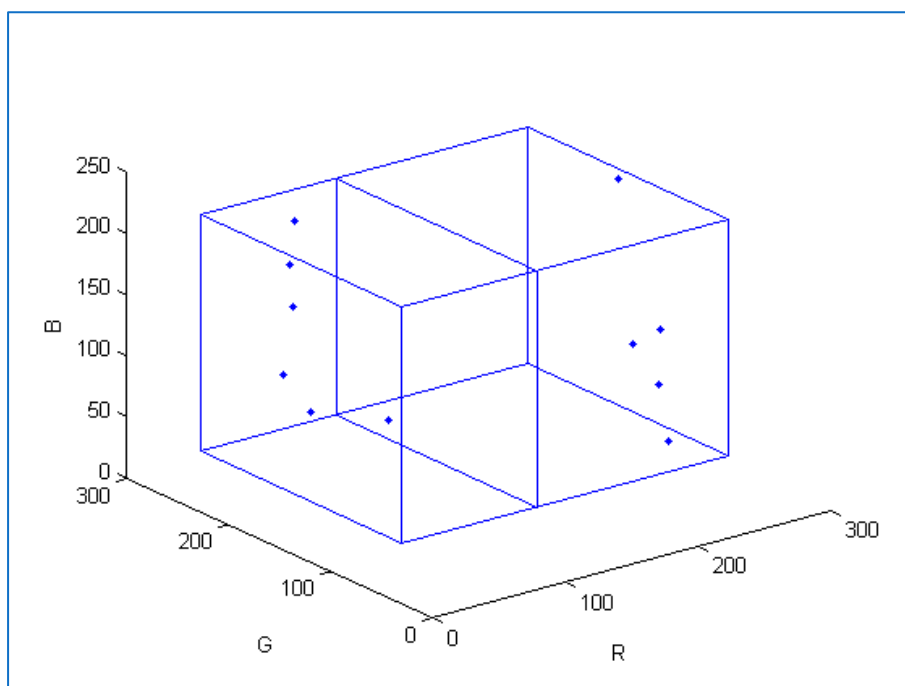


Figura 2.1.14: Paralelepípedo de la Figura 2.1.13 dividido según el algoritmo *Heckberts median-cut color quantization algorithm*

Como se había explicado, de esta forma cada uno de los dos nuevos paralelepípedos contiene el mismo número de puntos.

Este proceso se repite de forma recursiva para los nuevos paralelepípedos hasta que se obtiene un determinado número de paralelepípedos que contienen aproximadamente los mismos colores, y para cada uno de ellos se sustituyen todos los colores encerrados por un color representativo de todos ellos [1].

Para una imagen de tipo *8-bit Color*, el máximo de paralelepípedos en que se divide el espacio colorimétrico, que se corresponde con el máximo de colores que se pueden describir en la tabla LUT, es de $2^8 = 256$. Sin embargo, ImageJ también permite elegir cualquier otro número de colores al que reducir la imagen a convertir, siempre que se encuentre entre 2 y 256, tal y como se muestra en la Figura 2.1.15:

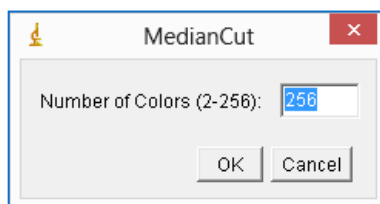


Figura 2.1.15. Menú contextual que permite elegir el número de colores que tendrá la nueva imagen de tipo *8-bit Color*

A continuación se muestra una comparación entre una imagen de tipo *RGB Color* y una imagen de tipo *8-bit Color* con diferentes números de colores en su paleta:

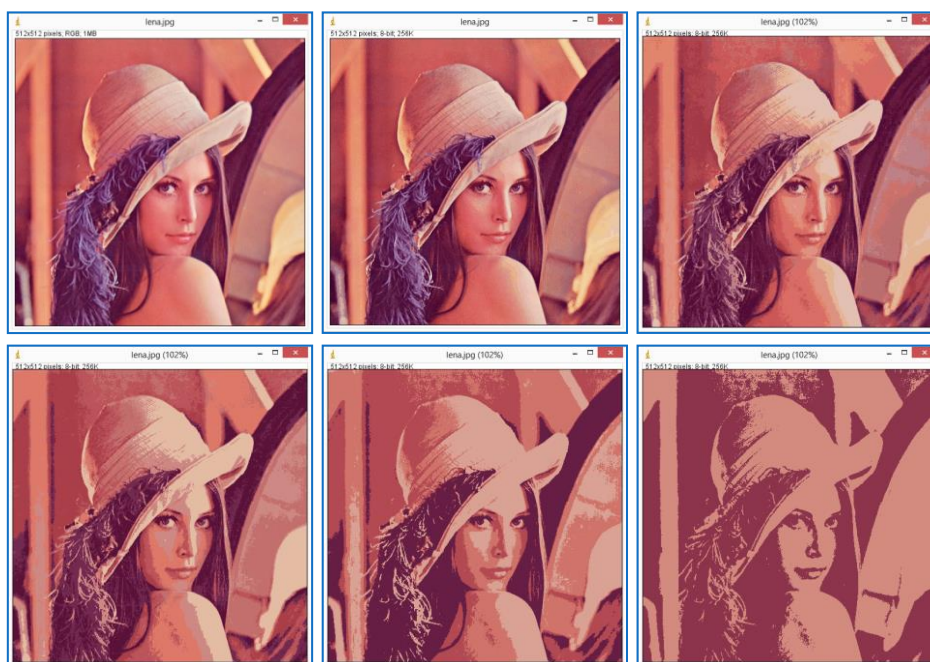


Figura 2.1.16. De izquierda a derecha y de arriba a abajo: imagen de tipo *RGB Color* original, imagen de tipo *8-bit Color* con 256 colores, con 16 colores, con 8 colores, con 4 colores y con 2 colores

En la Figura 2.1.16 se aprecia la pérdida de calidad que sufre la imagen conforme disminuye el número de colores que hay en su tabla LUT hasta llegar a la situación límite en la que sólo hay dos colores, uno representativo de la mitad de los colores originales, y otro representativo de la otra mitad.

Sin embargo, con un número suficientemente elevado de colores es posible conseguir mediante el algoritmo de *Heckbert* una representación fidedigna de la imagen original, como ocurre con las primeras imágenes de la fila superior de la Figura 2.1.16. En este caso es necesario acudir al *zoom* para distinguir las imperfecciones que deja la conversión, que se ponen de manifiesto en la Figura 2.1.17. En ella se aprecia que, en las zonas con píxeles que cuyo color varía de forma muy gradual, aparecen manchas uniformes porque la reducción de colores ha obligado a representar todos aquellos colores ligeramente diferentes con un mismo color. Estos problemas aparecen con imágenes reales, que tienen una gran gama de colores, entre los que puede haber colores que sólo se diferencian en pequeños matices, pero es más difícil que ocurra con imágenes sintéticas (como, por ejemplo, dibujos animados), porque tienen menos colores, zonas muy uniformes,... Esto hace que el tipo *8-bit Color* sea muy adecuado para estas imágenes, que en muchos casos se pueden representar sin ningún error con los 256 colores que caben en la tabla LUT.

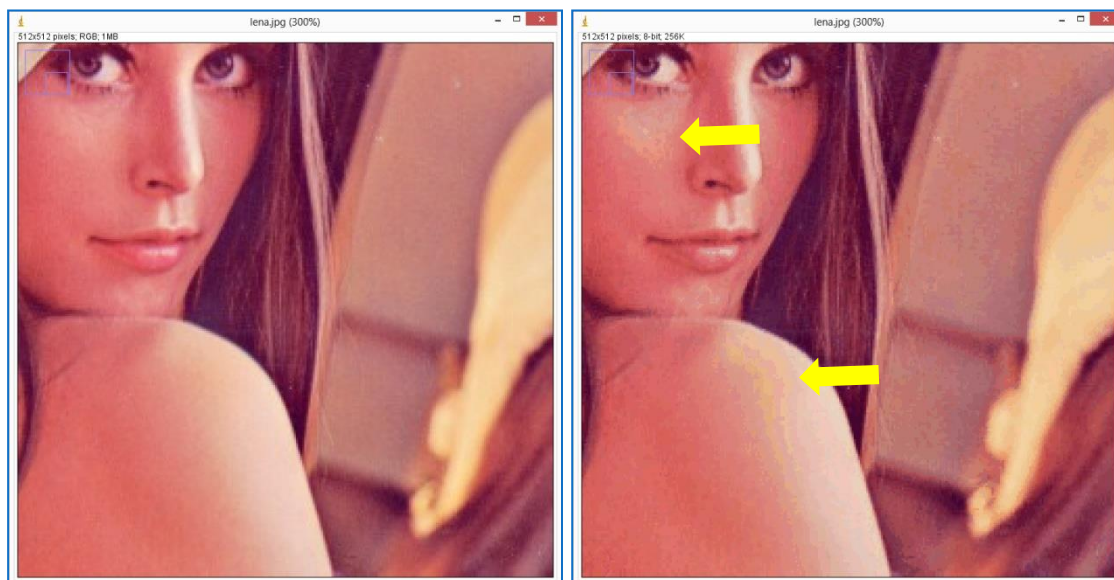


Figura 2.1.17. Comparación entre la imagen de tipo *RGB Color* original (izquierda) y la imagen de tipo *8-bit Color* final (derecha). En amarillo se señalan las imperfecciones que genera la conversión.

A pesar de estos defectos, la conversión a *8-bit Color* tiene la ventaja de reducir notablemente el tamaño del archivo que guarda la imagen: si antes se necesitaban $3 \cdot 8 = 24$ bits para describir el color de un píxel ahora basta con 8 bits y una mínima cantidad de información para guardar la tabla LUT.

2.1.5. RGB Color

Como ya se ha comentado, las imágenes de tipo *RGB Color* son imágenes a color en las que el color de cada píxel se describe en función de los niveles de sus tres componentes RGB. Es posible convertir una imagen a *RGB Color* desde cualquier otro tipo de imagen, e incluso en algunos casos se pueden convertir pilas de imágenes (ver Tabla 2.1.3). Los detalles de las conversiones se analizan a continuación:

■ Imagen original de tipo *8-bit*. El hecho de convertir una imagen de tipo *8-bit*, que es una imagen en escala de grises, a una imagen de tipo *RGB Color* no significa que se dote a la imagen original de color. Esta conversión sólo afecta al tipo de la imagen: si antes para cada píxel se describía su nivel de gris, una vez realizada la conversión a cada píxel se le asocian tres valores, los de sus componentes RGB. Sin embargo, el píxel sigue sin tener color, porque los tres valores de sus componentes RGB son el mismo (y ésta es la definición de gris), y de hecho coinciden con el nivel de gris único de la imagen original. Así que, visualmente, la imagen sigue siendo la misma una vez convertida. Esto se ilustra en la Figura 2.1.18, en la que no se aprecian cambios en la imagen. Sí se observa que un píxel de ejemplo, el de coordenadas $x=82$ e $y=68$, pasa de tener un nivel de gris de 107 a tener unas componentes $R=G=B=107$, lo cual significa lo mismo. Hay que reparar en otra consecuencia, y es que el hecho de describir la misma información con el triple de bits hace que la nueva imagen ocupe mucho más espacio, como también se puede ver en la figura. Es evidente que hay mucha información redundante y que esta conversión, al menos a priori, no es eficiente.

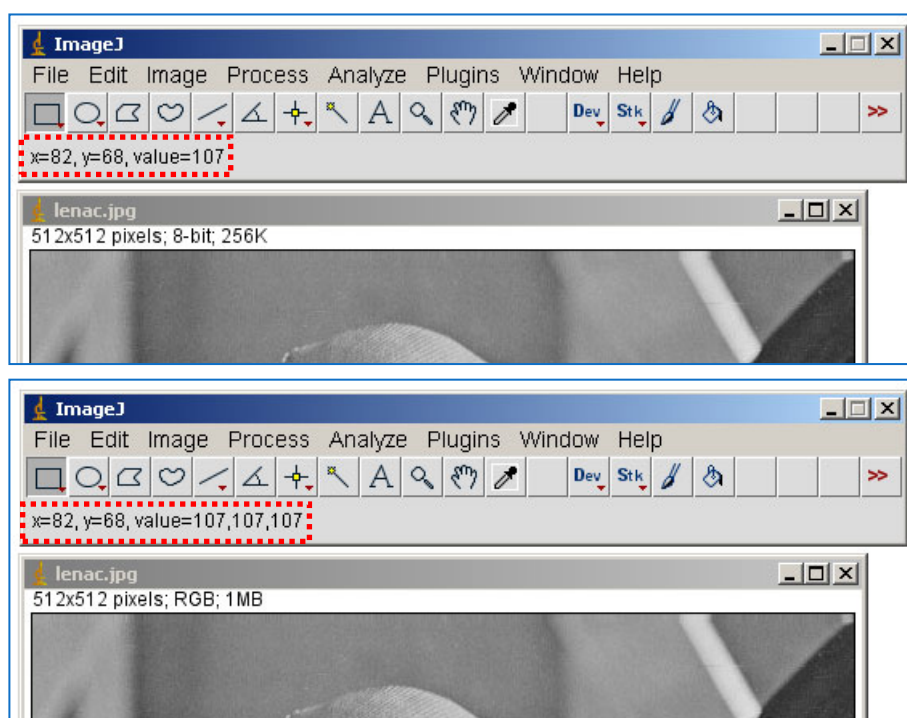


Figura 2.1.18. Arriba: imagen de tipo *8-bit*. Abajo: imagen anterior convertida a tipo *RGB Color*. En ambos casos, en la barra de estado se destacan en rojo los valores asociados al píxel de coordenadas $x=82$ e $y=68$

- Imagen original de tipo *16-bit*. En este caso sucede lo mismo que en el anterior. La única diferencia es que hay muchos más niveles de gris en la imagen original ($2^{16} = 65536$) que niveles de cada componente RGB en la imagen destino ($2^8 = 256$). Esto se soluciona haciendo una correspondencia entre los márgenes dinámicos de las imágenes originales y convertidas según la ecuación (2.1.1). Pero una vez solventada esta diferencia, vuelve a suceder lo mismo: se replica el nivel de gris de los píxeles de la

imagen original en los niveles de brillo de las tres componentes de los píxeles de la imagen convertida, que sigue siendo por tanto una imagen en escala de grises. En este caso, la ineficiencia de la conversión es aún mayor: además de aumentar de tamaño (porque se pasa de usar 16 bits por píxel a usar 32 bits por píxel, lo que significa el doble de tamaño), se pierde precisión porque, hay muchos menos niveles de gris (256 frente a 65536). La Figura 2.1.19 es un ejemplo de esta conversión. En ella se aprecia que el píxel destacado, que tenía un nivel de gris de 232, pasa a tener unas componentes R=G=B=6. Teniendo en cuenta que la imagen m51.tif tiene un nivel de gris máximo de 10106 y nivel mínimo de 0, estos nuevos valores de las componentes RGB concuerdan con lo explicado hasta ahora, puesto que según la ecuación (2.1.1),

$$v = \frac{255}{10106}(232 - 10106) + 255 \approx 5,854 \approx 6 \quad (2.1.8)$$

Por otro lado, la figura también muestra que después de la conversión el tamaño de la imagen aumenta hasta el doble del original (de 319 KB a 638 KB).

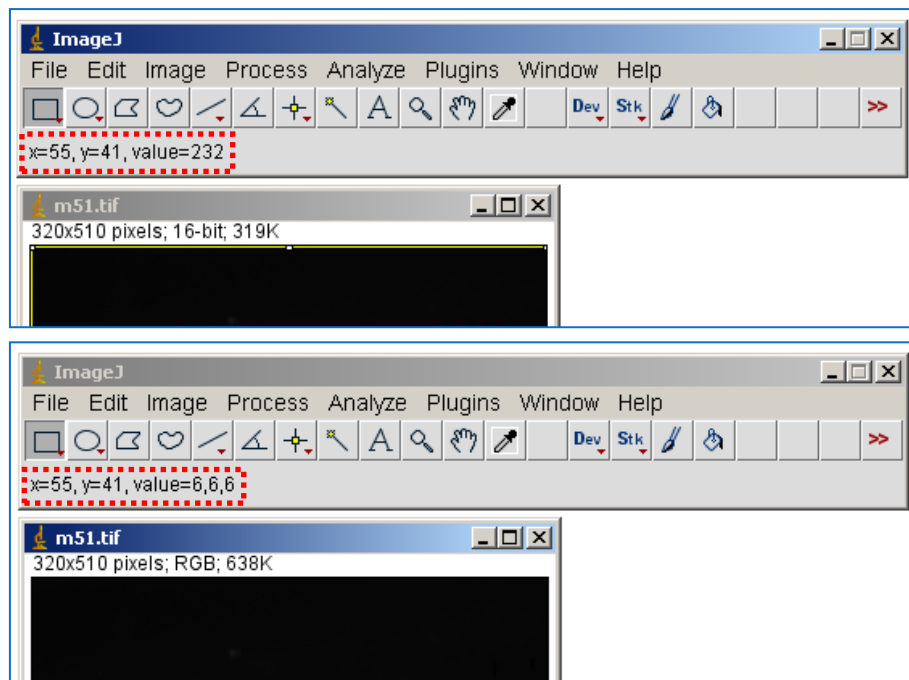


Figura 2.1.19. Arriba: imagen de tipo *16-bit*. Abajo: imagen anterior convertida a tipo *RGB Color*. En ambos casos, en la barra de estado se destacan en rojo los valores asociados al píxel de coordenadas $x=55$ e $y=41$

- Imagen original de tipo *32-bit*. Este caso es análogo al anterior.
- Imagen original de tipo *8-bit Color*. En esta ocasión se trata de una conversión a *RGB Color* desde una imagen que, aunque indexada, ya tenía color. La conversión consiste, sencillamente, en asociar a cada píxel las componentes RGB registradas en la entrada de la LUT a la que apuntaba su índice. Esto se comprueba en la Figura 2.1.20. El píxel de coordenadas $x=55$ e $y=28$ se describía en la imagen original

con un índice, el 28, que en la tabla LUT apuntaba a una entrada donde se describía el color R=35, G=0, B=144. En la imagen convertida, el mismo píxel se describe directamente con las componentes R=35, G=0 y B=144. Una vez más, esta conversión resulta un tanto ineficiente, puesto que, si bien la imagen convertida ocupa lo mismo que ocuparía una imagen capaz de contener 2^{24} colores diferentes, en realidad puede llegar a contener tan sólo 256 debido al mapeo directo que se hace desde la imagen tipo *8-bit Color*.

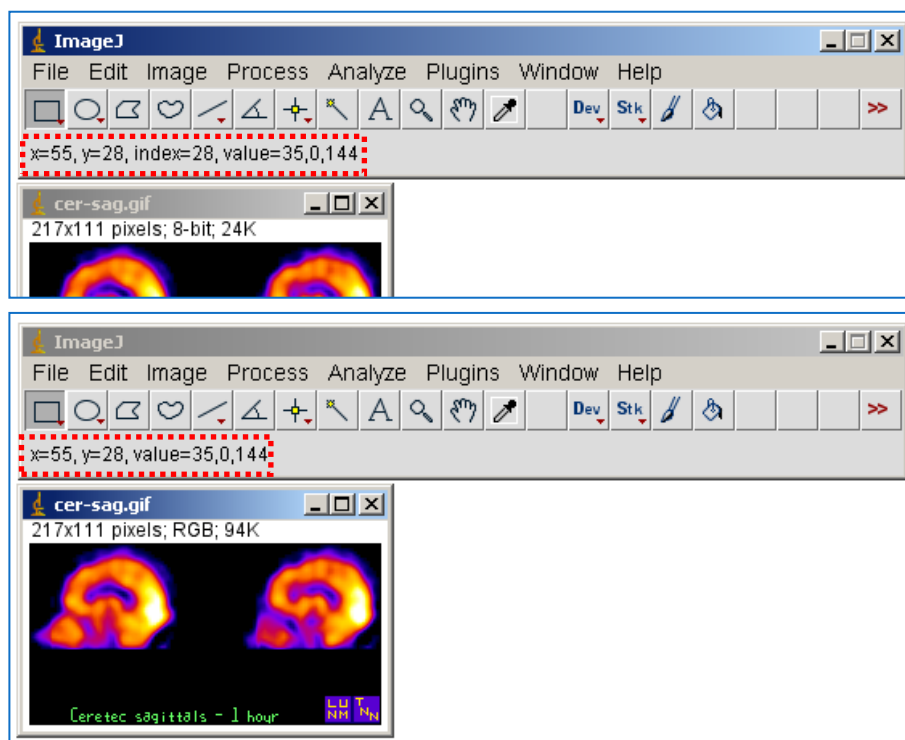


Figura 2.1.20. Arriba: imagen de tipo *8-bit Color*. Abajo: imagen anterior convertida a tipo *RGB Color*. En ambos casos, en la barra de estado se destacan en rojo los valores asociados al píxel de coordenadas $x=55$ e $y=28$

- Imagen original de tipo *RGB Stack*. Este tipo de imagen es una pila (una serie de imágenes que se muestran apiladas en la misma ventana) de 3 imágenes que se corresponden con los planos R, G y B de una imagen *RGB Color*. Estos planos, que se representan en escala de grises haciendo corresponder los niveles de la componente en cuestión con niveles de gris, muestran, por separado, la cantidad de rojo, verde y azul que hay en cada parte de la imagen. Por ejemplo, en la Figura 2.1.21, la primera imagen representa la cantidad de rojo que habría en la imagen si fuera de tipo *RGB Color*. Por ello, hay niveles altos de gris (próximos al blanco) en la piel, que tiene una gran cantidad de rojo, y sin embargo los niveles de gris son muy bajos (próximos al negro) en el adorno del sombrero, que es morado (ver Figura 2.1.16). La conversión desde este tipo de imagen al tipo *RGB Color* es sencilla. Para cada píxel se hace lo siguiente: a la nueva componente R se le hace corresponder el nivel de gris de la primera capa de la pila (el plano R); a la componente G, el nivel de gris de la segunda capa; y a la nueva componente B, la

tercera capa de la pila. Así se pasa de tener la información de rojo, verde y azul de la imagen por separado a tener toda la información cromática junta. En esta ocasión no aumenta el tamaño del resultado: se pasa de tener 3 imágenes de $m \times n$ píxeles cada una en las que se describe un píxel con 8 bits, a tener una sola imagen de tamaño $m \times n$ en la que se describen los píxeles con 24 bits ($3 \cdot m \times n \cdot 8 = m \times n \cdot 24$).

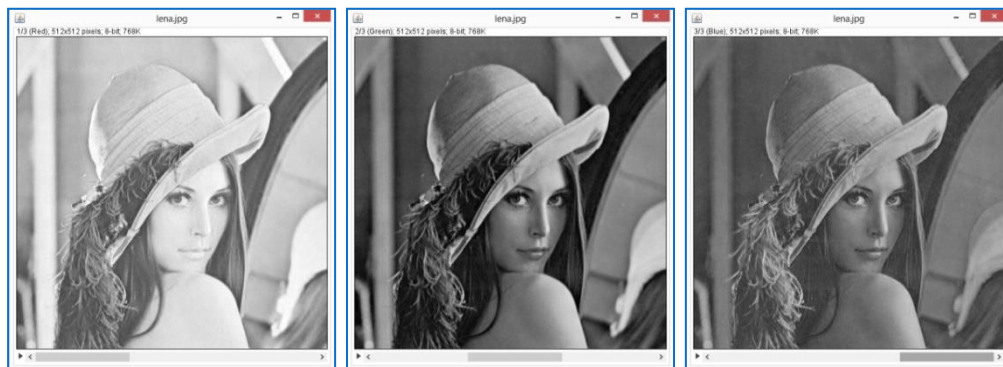


Figura 2.1.21. Capas de una imagen de tipo *RGB Stack* que se corresponden con los planos R, G y B (de izquierda a derecha) de una imagen de tipo *RGB Color*

- Imagen original de tipo *HSB Stack*. Este tipo de imagen es una pila en la que se encuentran separadas las componentes HSB (*Hue, Saturation y Brightness*, o Tono, Saturación y Brillo) de una imagen en color. Al igual que las componentes RGB, las componentes HSB permiten describir la información de brillo y color de una imagen, pero no se corresponden con sus colores primarios, sino con el tono, la saturación y el brillo de la imagen.

Si en el espacio de color RGB (Figura 2.1.12) se cortara el cubo de color con un plano $R=G=B=1$ se obtendría un triángulo como el de la Figura 2.1.22, que es el plano de trabajo en el que se describen las componentes HSB. El tono de los colores es la característica por la cual somos capaces de distinguir los rojos de los verdes, los azules de los amarillos, etc. En la Figura 2.1.22, el tono se corresponde con las diferentes zonas del plano. Esta diferencia se describe matemáticamente calculando los grados que separan la línea que une el centro del triángulo con la zona en la que está el color de la línea que une el centro con el rojo (elegido así por convenio). Así, los tonos rojos están a 0° , los azules a 120° , los verdes a 240° ,... Como las componentes HSB también se describen cada una con 8 bits, para representar el tono se escala el margen $0^\circ - 360^\circ$ y se traslada al margen $255 - 0$ (es decir, el negro se corresponde con los 360° , y por tanto con los rojos). La saturación de un color se refiere a su pureza. Un color con la misma tonalidad puede ser más o menos puro; es decir, sobre una misma línea que una el centro del triángulo con un punto de su perímetro (en una misma zona tonal), el color se puede encontrar más cerca del borde (más saturado) o más cerca del centro (menos saturado, más gris). En el plano S de las componentes HSB, un nivel 0 representa un gris (en el centro) y un nivel 255 representa un color totalmente saturado

(en el borde). Y por último, el brillo describe la distancia al plano de la Figura 2.1.22; cuanto más lejos del origen de coordenadas y más cerca del vértice blanco del cubo de la Figura 2.1.12 esté el color, más brillo tendrá.

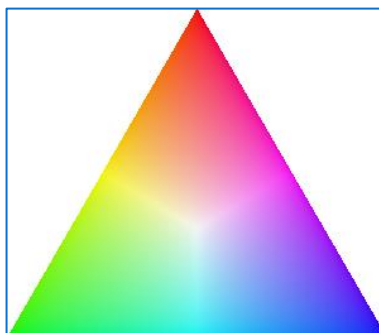


Figura 2.1.22. Plano de trabajo para las componentes HSB

En la Figura 2.1.23 se muestra una imagen *RGB Color* y sus componentes HSB separadas y en escala de grises.

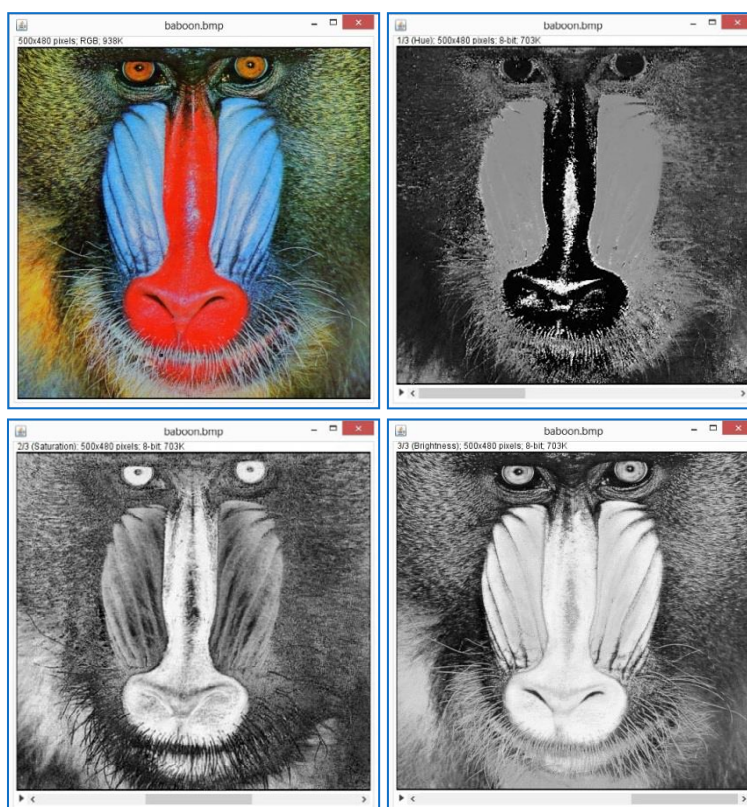


Figura 2.1.23. De izquierda a derecha y de arriba a abajo: imagen de tipo *RGB Color* original, tono, saturación y brillo

Se puede apreciar que, en la capa correspondiente al tono, el morro del babuino tiene partes muy negras y muy blancas, y esto concuerda con lo explicado anteriormente, porque el morro del babuino es rojo, que es una zona tonal a la que corresponden ángulos cercanos a $0^\circ=360^\circ$. Sin embargo, a la parte azul del morro, que es una zona tonal que se encuentra en torno a los 120° , le corresponde un nivel de gris más

intermedio. También se comprueba lo que se ha explicado en cuanto a la saturación: la parte roja del morro, que es de un color muy puro, aparece casi completamente blanca en la capa de saturación. Por el contrario, los pelos que tiene el babuino al final del morro son grises, sin ninguna saturación, y por eso aparecen como negros en la capa de saturación.

Para llevar a cabo la conversión de una imagen *HSB Stack* a una imagen *RGB Color* es necesario acudir a las ecuaciones que relacionan las componentes RGB y las HSB, que son las siguientes (en ellas se supone que H varía entre 0 y 360, que S varía entre 0 y 1, y que B varía entre 0 y 255):

$$\begin{aligned}
 & \blacksquare H_i = \left[\frac{H}{60} \right]_{mod 6} \\
 & \blacksquare f = \frac{H}{60} - H_i \\
 & \blacksquare p = B(1 - S) \\
 & \blacksquare q = B(1 - f \cdot S) \\
 & \blacksquare t = B(1 - (1 - f)S) \\
 & \hspace{15em} (2.1.9) \\
 & \blacksquare H_i = 0 \Rightarrow R = B, G = t, Blue = p \\
 & \blacksquare H_i = 1 \Rightarrow R = q, G = B, Blue = p \\
 & \blacksquare H_i = 2 \Rightarrow R = p, G = B, Blue = t \\
 & \blacksquare H_i = 3 \Rightarrow R = p, G = q, Blue = B \\
 & \blacksquare H_i = 4 \Rightarrow R = t, G = p, Blue = B \\
 & \blacksquare H_i = 5 \Rightarrow R = B, G = p, Blue = q
 \end{aligned}$$

Para obtener las componentes RGB de cada píxel, se aplican las ecuaciones recogidas en el conjunto (2.1.9) obteniendo los datos de entrada, H, S y B, de los píxeles con las mismas coordenadas en las diferentes capas de la pila.

2.1.6. RGB Stack

Como ya se ha avanzado, las imágenes de tipo *RGB Stack* son pilas de 3 imágenes que se corresponden con los planos R, G y B de una imagen *RGB Color*. En ImageJ sólo es posible realizar una conversión a este tipo de imagen desde imágenes *RGB Color* (y además sólo es posible aplicarlo a imágenes sencillas, y no a pilas). Para ello se crean tres nuevas imágenes de tipo *8-bit* de la misma dimensión que la original, y sus niveles de gris pasan a ser los niveles que tenían las componentes. En la Figura 2.1.24 se aprecia esto: en la imagen superior (de tipo *RGB Color*) el píxel de coordenadas x=56 e y=52 tiene componentes R=228, G=118 y B=95. Esta imagen se transforma en una pila *RGB Stack*, y en la parte inferior de la figura se muestra su primera capa, que corresponde a la componente R; en ella se observa que el píxel de coordenadas x=56 e y=52 tiene un nivel de gris de 228, que coincide con el valor de la componente R en la imagen original.

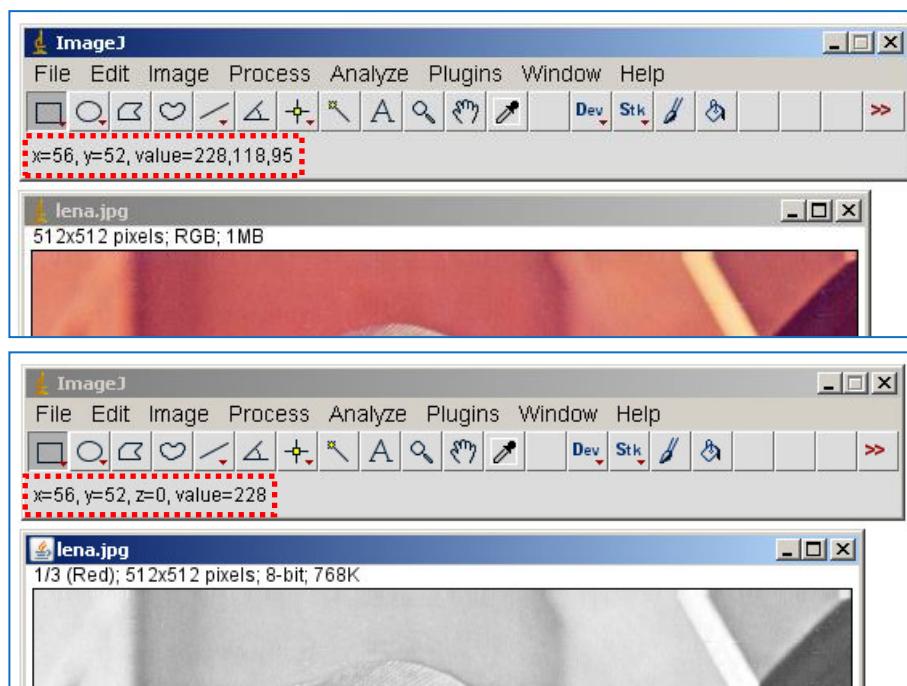


Figura 2.1.24. Arriba: imagen de tipo *RGB Color*. Abajo: plano R de la pila *RGB Stack* a la que se convirtió la imagen anterior. En ambos casos, en rojo se destaca la información relativa al píxel de coordenadas $x=56$ e $y=52$

2.1.7. HSB Stack

Para terminar, la opción *HSB Stack* permite convertir imágenes *RGB Color* a pilas con capas correspondientes al tono, la saturación y el brillo de la imagen original. Este caso es análogo al anterior, con la diferencia de que hay que hacer una conversión previa de componentes RGB a componentes HSB, para lo cual se usan ecuaciones inversas a las que se agrupan en (2.1.9).

En la Tabla 2.1.3 se muestra un resumen de las posibles conversiones que se pueden llevar a cabo en el submenú *Type*:

Tabla 2.1.3. Resumen de las posibilidades de conversión en el submenú *Type*

De ↓	A →	8-bit	16-bit	32-bit	8-bit Color	RGB Color	RGB Stack	HSB Stack
8-bit		x	Imagen Pila	Imagen Pila	x	Imagen Pila	x	x
16-bit		Imagen Pila	x	Imagen Pila	x	Imagen Pila	x	x
32-bit		Imagen Pila	Imagen Pila	x	x	Imagen Pila	x	x
8-bit Color		Imagen Pila	x	x	x	Imagen	x	x
RGB Color		Imagen Pila	Imagen	Imagen	Imagen Pila	x	Imagen Pila	Imagen Pila
RGB Stack		x	x	x	x	Imagen	x	
HSB Stack		x	x	x	x	Imagen	x	x

2.2. Submenú Adjust

El submenú *Adjust* (Ajustar) permite hacer ajustes minuciosos sobre el brillo, el contraste, el color y el tamaño de las imágenes, y también proporciona herramientas para umbralizarlas.

2.2.1. Brightness/Contrast

Esta opción despliega un menú contextual desde el que se puede modificar interactivamente el brillo y el contraste de la imagen activa. Aunque ya se ha hablado sobre el brillo, no se ha introducido aún el concepto de contraste. Esta magnitud se refiere a la diferencia de brillo que se puede encontrar en la imagen: cuanto mayor sea la diferencia de brillo entre los píxeles más brillantes y los menos brillantes de una imagen, más contraste tiene ésta.

Este menú contextual se compone de los siguientes elementos:

- **Histograma.** Aparece en la parte superior del menú. El histograma de una imagen es un gráfico en el que el eje de abscisas alberga ordenados los diferentes niveles de brillo que podría tomar un píxel en la imagen y el eje

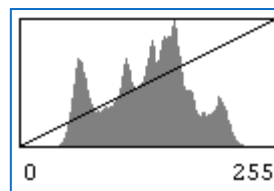


Figura 2.2.1. Histograma de lena.jpg

de ordenadas contiene el número de píxeles de la imagen que han tomado cada nivel de brillo. El histograma da una idea de la distribución del brillo en una imagen. Un histograma en el que el contenido se acumule en la izquierda, que normalmente se asocia a los niveles de brillo más bajos, corresponderá a una imagen oscura, y viceversa. Además, sobre el histograma se imprime una línea recta que muestra cómo se mapean los posibles niveles de brillo que puede tomar la imagen (en el eje de abscisas) en los 256 niveles de brillo que se representan en la pantalla. En el caso de la Figura 2.2.1, los niveles de brillo que se pueden representar en la imagen también son 256, así que el hecho de que la línea recta sea la bisectriz del cuadrante significa que el nivel de brillo 1 se representará en la pantalla como el nivel de brillo 1, que el nivel de brillo 2 se representará en la pantalla como el nivel de brillo 2, etc. Por último, bajo el histograma aparecen dos números, que se corresponden con los niveles de brillo mínimo y máximo mapeados en los niveles de brillo de la pantalla.

- **Barras deslizantes *Minimum* y *Maximum*.** Con estos controles se pueden modificar los niveles de brillo de la imagen mínimo y máximo que se mapean en los niveles de brillo de la pantalla. Esto equivale a desplazar horizontalmente los extremos de la línea de mapeo y, evidentemente, afecta al brillo y al contraste de la imagen que se muestra en pantalla. El brillo aumenta cada vez que se disminuyen los valores del mínimo y del máximo. Si el nivel de brillo mínimo que se mapea es muy alto, puede que haya una gran cantidad de niveles (los inferiores al mínimo) a los que directamente se les haga corresponder el nivel 0, y en cualquier caso a los primeros niveles mapeados se les hace corresponder niveles de salida

muy bajos, así que la imagen será oscura; de ahí que el brillo de la imagen aumente al acercar dicho mínimo al 0. La explicación es análoga para el caso del valor máximo: si el nivel de brillo máximo que se mapea es muy bajo, puede que haya una gran cantidad de niveles (los superiores al máximo) a los que directamente se les haga corresponder el nivel 255, y en cualquier caso a los últimos niveles mapeados se les hace corresponder niveles de salida muy altos, así que la imagen será clara; de ahí que el brillo de la imagen aumente al disminuir el máximo. Por otro lado, el contraste aumenta cada vez que aumenta el mínimo o disminuye el máximo. Esto ocurre porque al aumentar el mínimo y disminuir el máximo las partes oscuras se hacen aún más oscuras y las partes claras se hacen aún más claras. Además, son muchos los píxeles que quedan fuera del intervalo que se mapea linealmente, y a todos ellos se les hace corresponder 0 o 255 directamente. Entre un píxel que tienen un nivel de brillo de 0 y otro que tiene un nivel de brillo de 255 existe la máxima diferencia de brillo posible en una imagen con 256 niveles, y por tanto, el contraste más alto.

- Barra deslizante *Brightness*. Este control modifica el brillo conjunto de toda la imagen que se muestra en la pantalla. Gráficamente, desplaza horizontalmente toda la línea de mapeo sin modificar su pendiente, pero desplazando con ella sus extremos y, por tanto, los niveles de brillo mínimo y máximo que se mapean en los niveles de brillo de la pantalla. Desde el punto de vista matemático, actuar sobre este control equivale a modificar el término independiente b de la ecuación lineal que describe el mapeo:

$$y = mx + b$$

$$\text{nivel mapeado} = \text{contraste} \cdot \text{nivel original} + \text{brillo} \quad (2.2.10)$$

Desplazar la línea de mapeo a la izquierda, que es lo mismo que aumentar el término independiente, equivale por tanto a aumentar el brillo de la imagen que se muestra en pantalla (y viceversa).

- Barra deslizante *Contrast*. Con esta barra deslizante se modifica el contraste total de la imagen que aparece en pantalla. A la hora de hacer el mapeo de los niveles de brillo de los píxeles que hay en la imagen a los que se muestran en pantalla, actuar sobre este control equivale a modificar la pendiente de la línea de mapeo, a la que se denota como m en la ecuación 2.2.10. Aumentar el contraste implica hacer más pronunciada la pendiente de la línea de mapeo, lo cual también afecta directamente a los niveles de brillo mínimo y máximo que se mapean en los niveles de brillo de la pantalla, que en este caso se aproximarían el uno al otro.
- Botón *Auto*. Este control permite optimizar automáticamente el brillo y el contraste de la imagen que se muestra en pantalla. Para ello se utiliza la técnica conocida como *Contrast Stretching* (literalmente, Estiramiento del Contraste). Esta técnica se basa en estirar el margen dinámico de la imagen original hasta hacerlo coincidir con el margen dinámico máximo disponible según el tipo de imagen (256 en

imágenes de tipo *8-bit*, 65536 en las imágenes de tipo *16-bit*,...). Se puede implementar mediante un sencillo escalamiento, tan sólo hay aplicar la siguiente expresión al nivel de brillo de cada píxel (la siguiente expresión es aplicable cuando el margen dinámico de destino es 0-255):

$$v = (u - \min) \left(\frac{255 - 0}{\max - \min} \right) + 0 \quad (2.2.11)$$

donde *min* es el nivel de brillo mínimo que se puede encontrar en la imagen original y *max* es el nivel de brillo máximo [2]. Sin embargo, esta transformación es muy sensible a valores *outliers* (niveles de brillo atípicos, muy brillantes o muy oscuros, que generalmente son la consecuencia de algún tipo de error), que afectan directamente a los parámetros *min* y *max* y por tanto modifican el escalamiento y generan peores resultados. Para evitar esto y mejorar esta operación se suele llevar a cabo de forma previa un análisis del histograma de la imagen y se descarta un pequeño porcentaje de píxeles, que en ImageJ es concretamente 0,35%. Estos píxeles corresponden a los extremos del histograma (para evitar niveles de brillo *outliers*) y en la imagen de salida aparecerán saturados.

Con todo esto se busca realizar un mapeo tal que todo el margen dinámico del que se dispone para mostrar la imagen se aproveche para representar el grueso de la información (todo salvo los pocos píxeles saturados). Esto se ilustra en la Figura 2.2.2.

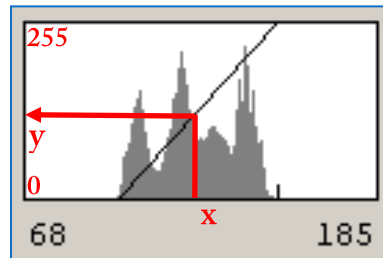


Figura 2.2.2. Histograma de la imagen Tulips.jpg con la línea de mapeo que se genera al pulsar el botón *Auto*

En la figura anterior se observa cómo, tras pulsar el botón *Auto*, se ajustan el *offset* y la pendiente de la línea de mapeo (es decir, el brillo y el contraste, respectivamente) para que la parte del margen dinámico de entrada que se mapea sea aquella que contiene el grueso de la información, que en este caso es la que se comprende entre los niveles de brillo 68 y 185. Seguramente hay unos pocos píxeles que toman niveles de brillo inferiores a 68 o superiores a 185 y que por lo tanto quedarán saturados, pero esto es necesario para que el algoritmo que subyace al proceso funcione de forma óptima. Con esta línea de mapeo, los píxeles que en la imagen original tengan un nivel de brillo menor o igual que 68 pasarán a tener un nivel de brillo 0 (serán negros) en la imagen que se muestra en pantalla, así como los píxeles originales con nivel de brillo de 185 o superior tendrán un nuevo nivel de brillo de 255 (serán blancos), y en general cualquier píxel con nivel de brillo *x* pasará a tener un nivel de brillo *y* en la imagen que aparece en

pantalla según el mapeo lineal que se muestra gráficamente en la Figura 2.2.2. A continuación se muestra la imagen original a la que corresponde el histograma de la figura anterior y la imagen con niveles de brillo optimizados según el mapeo que también se describía en la figura anterior:

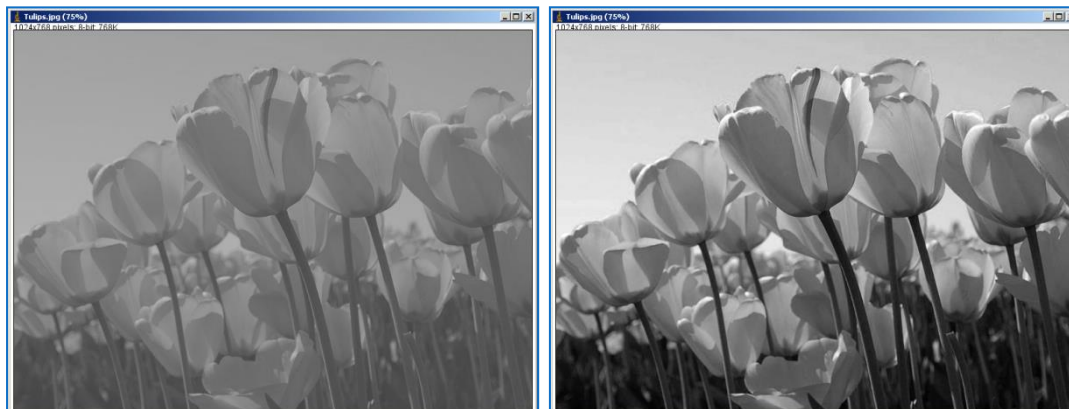


Figura 2.2.3. Izquierda: imagen original. Derecha: imagen original con el brillo y el contraste optimizados automáticamente

En la Figura 2.2.3 se puede apreciar la gran diferencia que existe entre una imagen cualquiera (izquierda) y la misma imagen con el brillo y el contraste optimizado (derecha). En esta última se aprovecha todo el margen dinámico del que se dispone para mostrar la imagen, de forma que los píxeles más oscuros que había en la imagen se muestran tan oscuros como es posible y lo mismo sucede con los píxeles más brillantes. De esta forma la visualización de la imagen resulta mucho más cómoda a la vista que antes, cuando los niveles de brillo eran tan parecidos que no resultaba tan fácil interpretar la información de la imagen, separar los objetos que aparecen en ella, etc.

Antes de pasar a describir el siguiente control hay que destacar la posibilidad de optimizar el brillo y el contraste de toda una imagen en base al análisis de una sola región de la imagen. Para ello sólo es necesario crear una selección de área con las herramientas de selección que se pueden encontrar en la barra de tareas de ImageJ y pulsar el botón *Auto*. De esta forma, el análisis previo se realiza sólo sobre el histograma de la región seleccionada, y a partir de él se extrae la regla de mapeo lineal que se aplica a los píxeles de toda la imagen.

- Botón *Reset*. Este botón resetea la función de mapeo a su estado por defecto, que es la bisectriz del cuadrante en el que se muestra el histograma:

$$y = 1 \cdot x + 0 = x \quad (2.2.12)$$

Al pulsar el botón la imagen que se muestra en pantalla también se resetea para cumplir con el mapeo anterior, lo que evidentemente significa que la imagen que se representa es la original, sin cambios de brillo o contraste.

- Botón *Set*. Cuando se pulsa este botón aparece el menú que se representa en la Figura 2.2.4. Este menú permite establecer de forma numérica los niveles de brillo mínimo y máximo que se mapean y se muestran en pantalla (*Minimum displayed value* y *Maximum displayed value*), lo cual puede resultar útil y más cómodo que un ajuste manual con las barras deslizantes *Minimum* y *Maximum* si, por ejemplo, se quisiera optimizar el brillo y el contraste de una imagen conocidos los valores numéricos de sus niveles mínimo y máximo de brillo.

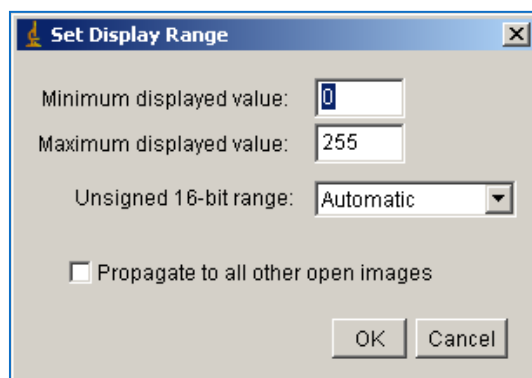


Figura 2.2.4. Menú contextual que se despliega al pulsar el botón *Set*

Además, el menú proporciona la lista desplegable *Unsigned 16-bit range*, en la que se puede elegir la forma de manejar el margen dinámico de imágenes de tipo *16-bit*. Se proporcionan las siguientes opciones:

- *Automatic*. Se selecciona de forma automática la parte del margen dinámico de la imagen de tipo *16-bit* (es decir, el subintervalo de niveles de brillo dentro del intervalo 0-65535) que se tiene en cuenta.
- *8-bit*. Sólo se tienen en cuenta los niveles de brillo que van desde 0 hasta 255 (como si se tratara de una imagen de tipo *8-bit*). El resto del margen dinámico, a partir de 256, se descarta.
- *10-bit*. Sólo se tienen en cuenta los niveles de brillo que van desde 0 hasta 1023 (como si se tratara de una imagen de un hipotético tipo *10-bit*). El resto del margen dinámico, a partir de 1024, se descarta.
- *12-bit*. Sólo se tienen en cuenta los niveles de brillo que van desde 0 hasta 4095 (como si se tratara de una imagen de un hipotético tipo *12-bit*). El resto del margen dinámico, a partir de 4096, se descarta.

- *15-bit*. Sólo se tienen en cuenta los niveles de brillo que van desde 0 hasta 32767 (como si se tratara de una imagen de un hipotético tipo *15-bit*). El resto del margen dinámico, a partir de 32768, se descarta.
- *16-bit*. Se tiene en cuenta todo el margen dinámico de la imagen tipo *16-bit*.

Para terminar, el menú ofrece la casilla *Propagate to all other open images*. Al marcarla, se aplican los cambios a todas las imágenes abiertas en la sesión actual de ImageJ.

- Botón *Apply*. Todos los controles del submenú *Brightness/Contrast* descritos hasta ahora afectan a la forma de mapear los niveles de brillo de la imagen original en los niveles de brillo de la imagen que se muestra en pantalla, pero salvo en el caso de las imágenes *RGB Color*, nunca se modifican los píxeles de la imagen original. En el resto de los casos, para que las modificaciones que se muestran en pantalla se guarden sobre los píxeles originales es necesario pulsar este botón. De hecho, al pulsar *Apply*, el histograma que se muestra pasa a ser el de la imagen ya modificada y la función de mapeo vuelve a tomar su forma por defecto (la bisectriz).

2.2.2. Window/Level

Esta opción tiene la misma función que la opción anterior: modificar el brillo y el contraste de la imagen activa. Sin embargo, mientras que la opción *Brightness/Contrast* permite ajustar la función de mapeo actuando sobre parámetros como los niveles de brillo mínimo y máximo (los extremos de la función), el brillo (el término independiente de la función) o el contraste (la pendiente de la función), esta opción modifica la función de mapeo en base a los conceptos de *window* (ventana) y *level* (nivel).

- Barra deslizante *Window*. Se refiere a la ventana que abarca todos los niveles de brillo que se mapean sin saturar, así que está delimitada por los niveles de brillo mínimo y máximo que se mapean en la imagen en pantalla. Llevando la barra deslizante hacia la derecha se hace la ventana cada vez más estrecha, reduciendo el número de píxeles que se mapean linealmente (los que se quedan dentro de la ventana) y aumentando el número de aquellos que se saturan a 0 y 255 (los que se quedan fuera de la ventana). Esto, como ya se ha explicado, aumenta el contraste de la imagen. Bajo la barra *Window* se muestra el ancho de la ventana en niveles de brillo, que puede ir desde 2 (una pendiente casi infinita que conlleva una imagen binaria) hasta un valor muy alto (una pendiente casi nula que conlleva una imagen uniforme en la que todos los píxeles tienen casi el mismo nivel de brillo). En la Figura 2.2.5 se aprecia la diferencia entre una imagen que se mapea en la pantalla con una ventana de 271 niveles de brillo de ancho y la misma imagen mapeada con sólo 24, que como se acaba de explicar tiene mayor contraste (de hecho es excesivo).

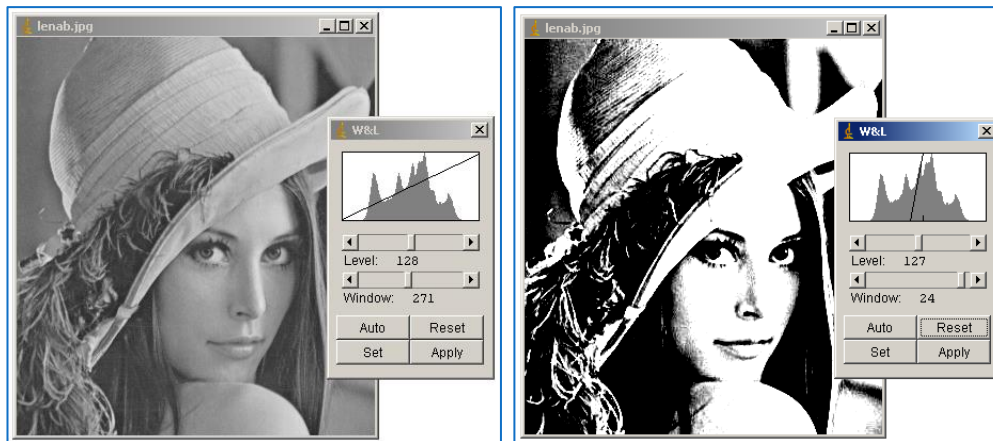


Figura 2.2.5. Diferencias en el contraste que se aprecian al actuar sobre el control *Window*

- Barra deslizante *Level*. Mueve la ventana sobre el histograma. Aumentar el nivel (desplazar la barra hacia la derecha) implica mover la ventana hacia la izquierda, y por tanto mapear niveles más bajos de la misma forma que antes se mapeaban niveles más altos, con lo que la imagen gana en nivel de brillo (de ahí el nombre de *Level*). Bajo la barra se muestra el nivel de brillo original sobre el que se centra la ventana. En la Figura 2.2.6 se aprecia la diferencia entre una imagen que se mapea en la pantalla con una ventana que ocupa la parte derecha del histograma y la misma imagen mapeada con la misma ventana desplazada a la parte izquierda del histograma.

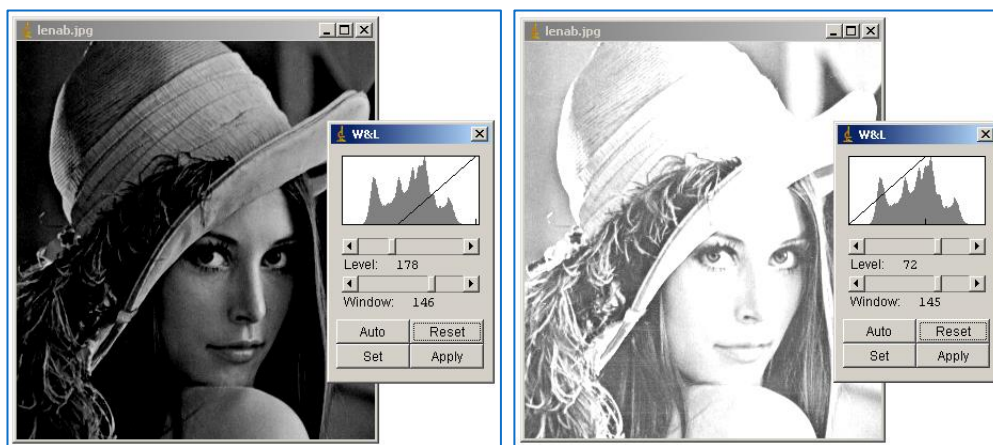


Figura 2.2.6. Diferencias en el brillo que se aprecian al actuar sobre el control *Level*

El resto de los botones (*Auto*, *Reset*, *Set* y *Apply*) son totalmente análogos a los que se describieron en el apartado anterior. Antes de pasar al siguiente apartado hay que señalar que se puede alternar entre las opciones *Brightness/Contrast* y *Window/Level* sin perder las modificaciones realizadas sobre la función de mapeo (si se elige la opción *Brightness/Contrast* se abre su menú contextual en el lugar del de *Window/Level*, que se cierra, y se continúa trabajando sobre la mismas función de mapeo).

2.2.3. Color Balance

Esta opción también está enfocada a manipular el brillo y el contraste, aunque a diferencia de las opciones anteriores, permite operar por separado sobre los planos RGB de una imagen *RGB Color*. También permite abrir imágenes indexadas y en escala de grises, pero en estos casos no aporta ninguna ventaja sobre las opciones anteriores; en el primer caso, porque el propio concepto de imagen indexada no permite trabajar sobre el histograma de esta forma, y en el segundo, porque sencillamente hay un solo plano, el del nivel de brillo. Al seleccionar esta opción aparece un menú contextual que es prácticamente igual al menú *Brightness/Contrast*; sólo difiere de él en que en lugar de la barra deslizante *Contrast* (para ajustarlo se utilizan las otras barras) aparece una lista desplegable que permite elegir el plano o el canal sobre el que se va a trabajar (en el panel superior se muestra el histograma correspondiente). Por lo demás, el funcionamiento es el mismo que se explicó en apartados previos. A continuación, en la Figura 2.2.7, se muestra un ejemplo del trabajo con esta opción.

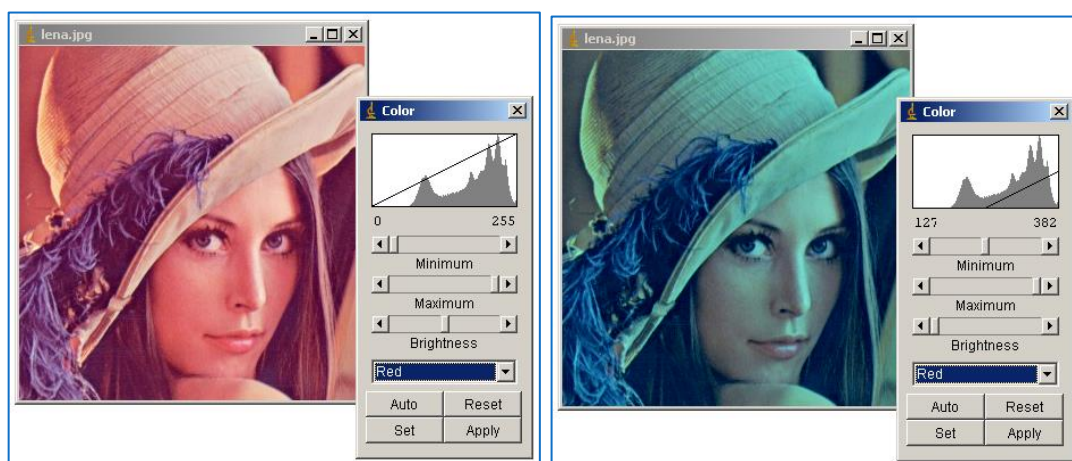


Figura 2.2.7. Izquierda: imagen *RGB Color* original. Derecha: imagen original con modificaciones el plano rojo

En la Figura 2.2.7 se aprecia a la izquierda una imagen de tipo *RGB Color* y el menú *Color Balance*. Se ha seleccionado el plano rojo y en el panel superior del menú se muestra su histograma, en el que se observa que en la imagen abundan los píxeles con niveles altos de rojo, puesto que todo el contenido del histograma se agrupa a la derecha. A la derecha aparece la misma imagen, pero la forma en la que se mapea su plano rojo en pantalla ha cambiado: la barra *Brightness*, que ahora actúa sobre el nivel de la componente roja, está totalmente a la izquierda, así que se ha quitado una gran carga de rojo a la imagen, lo cual se aprecia fácilmente al comparar ambas imágenes.

A pesar de que no incluye muchos más parámetros que los menús anteriores, se trata de una opción bastante versátil; operando con los controles se puede conseguir, por ejemplo, dejar de mapear en pantalla los planos verde y azul, y extraer de esta forma el plano rojo coloreado de rojo, como se muestra en la Figura 2.2.8:

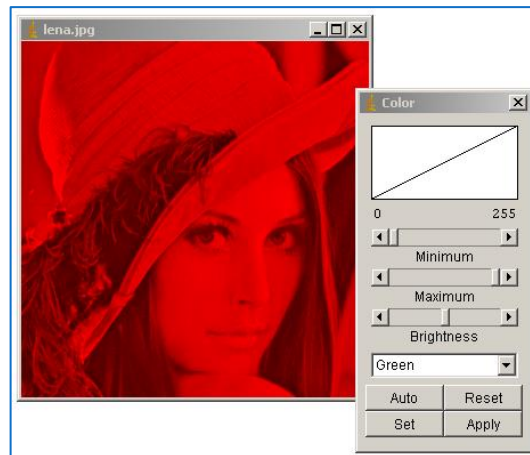


Figura 2.2.8. Imagen de tipo *RGB Color* en la que no se mapean en pantalla los planos verde y azul, y que por tanto coincide con su plano rojo coloreado de rojo

Hay que señalar que para que los cambios que se realicen sobre las funciones de mapeo se conserven al cambiar de plano o canal hay que pulsar sobre el botón *Apply*. Y además, como ocurría con la opción anterior, se puede alternar entre esta opción y la opción *Brightness/Contrast*.

2.2.4. Threshold

La opción *Threshold* (Umbral o Umbralizar) despliega un menú contextual que permite umbralizar imágenes en escala de grises, es decir, segmentarlas en función de sus niveles de gris para separar las características de interés del resto de la imagen (al cual se refiere ImageJ como *Background*, fondo). Para ello, en el menú se ofrecen los siguientes controles:

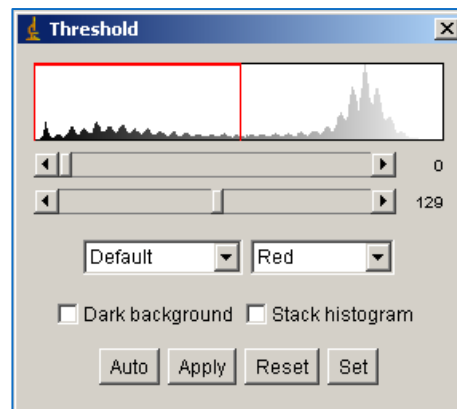


Figura 2.2.9. Menú contextual correspondiente a la opción *Threshold*

Barras deslizantes de los umbrales. Permiten ajustar los umbrales inferior (mediante la barra de arriba) y superior (mediante la barra de abajo). A su derecha se muestra el nivel de gris sobre el que está situado cada umbral. Estos umbrales acotan los niveles de gris que toman las características de interés de la imagen y que, por tanto, se van a segmentar. En la Figura 2.2.10 hay una imagen en escala de grises umbralizada con diferentes umbrales.

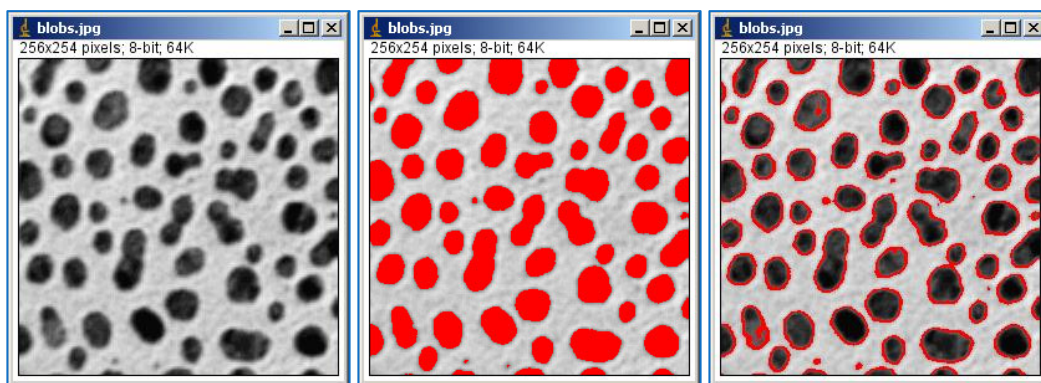


Figura 2.2.10. Izquierda: imagen original. Centro: imagen umbralizada desde 0 a 140. Derecha: imagen umbralizada desde 101 a 170

En el centro se muestra la misma imagen umbralizada con el intervalo 0-140 segmentado en rojo. Dicho intervalo corresponde a las partículas, que son oscuras, al contrario que el fondo, que es más claro. A la derecha aparece la misma imagen segmentada con otros umbrales: desde 101 a 170. Entre estos dos valores se encuentran los niveles de gris de la mayoría de los bordes, y por ello aparecen destacados en rojo.

- Lista desplegable del método de umbralización. Permite elegir entre diferentes algoritmos de umbralización automática basada en el histograma [3]. Unos trabajan mejor con ciertas imágenes que otros, pero hay que tener en cuenta que, en cualquier caso, no son efectivos si se aplican a imágenes iluminadas de forma muy desigual.
- Lista desplegable de la representación. Permite elegir la forma de representar la imagen segmentada. Existen las siguientes posibilidades:
 - *Red*. Los niveles de gris umbralizados (acotados por los umbrales mínimo y máximo) se representan en rojo y en el fondo se mantienen los niveles de gris originales (ver Figura 2.2.10)
 - *B&W. Black and White*. Los niveles umbralizados se representan en negro, mientras que todos los píxeles del fondo pasan a ser blancos (Figura 2.2.11, izquierda)
 - *Over/Under*. Permite cambiar el modo de trabajo y umbralizar la imagen por partida doble: desde el negro hasta el umbral inferior y desde el umbral superior hasta el blanco, dejando los niveles de gris que hay entre ambos umbrales como fondo. Los píxeles cuyos niveles estaban en el intervalo inferior pasan a ser azules y los que estaban en el intervalo superior se muestran en verde, mientras que los píxeles del fondo conservan su nivel de gris (Figura 2.2.11, derecha). Los colores con los que se muestran las zonas segmentadas se pueden cambiar con un *macro* y usando los métodos *ImageProcessor.setOverColor()* e *ImageProcessor.setUnderColor()* (ver el apartado 5.1 Submenú *Macros*).

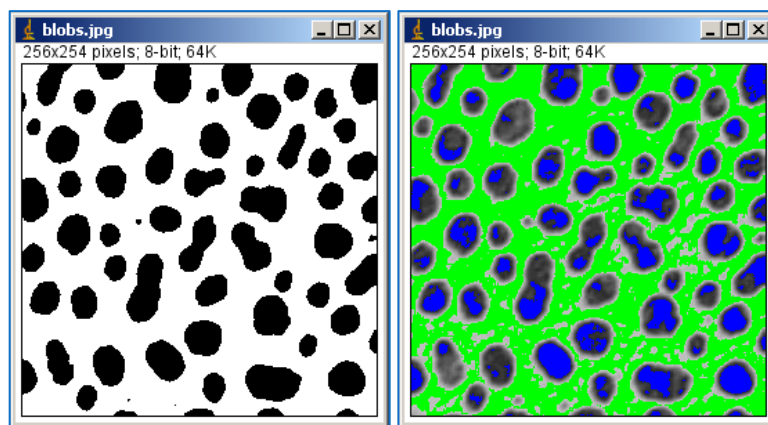


Figura 2.2.11. Izquierda: imagen umbralizada y representada según *B&W*. Derecha: imagen umbralizada y representada según *Over/Under*

- *Dark background* (fondo negro). Su efecto es el de intercambiar la forma de representar las partes de interés de la imagen por la forma de representar el fondo. Así, si el modo de representación es *Red*, los píxeles con un nivel de brillo contenido en el intervalo delimitado por los umbrales mantienen su nivel, pero los demás pasan a ser rojos.
- *Stack histogram*. Si se está umbralizando una pila de imágenes y se marca esta opción, ImageJ tiene en cuenta el histograma conjunto de todas las imágenes de la pila y luego ejecuta el algoritmo de umbralización por igual para todas ellas.
- *Auto*. Este botón permite hacer una umbralización automática con el algoritmo de umbralización elegido en la correspondiente lista desplegable.
- *Apply*. Hace efectiva la umbralización reescribiendo los niveles de brillo de los píxeles en el archivo original de la imagen, dejando todos los píxeles de la región de interés y del fondo negros y blancos respectivamente.
- *Reset*. Restituye la imagen a su estado original
- *Set*. Permite definir numéricamente los umbrales.

Esta opción, en combinación con otras herramientas de ImageJ, permite llevar a cabo operaciones muy útiles. Por ejemplo, en la Figura 2.2.12 se umbraliza la imagen de la izquierda, se segmentan las gotas y se destacan en rojo uniforme, y se usa la herramienta *Wand tracing tool* (que delimita zonas de la imagen con el mismo color que el píxel en el que se aplica) para seleccionar una gota. Después, con la opción *Measure* del menú *Analyze* se obtiene un resumen de las características de esa partícula en concreto: el área en píxeles que ocupa, sus niveles máximo y mínimo y la media de nivel de brillo.

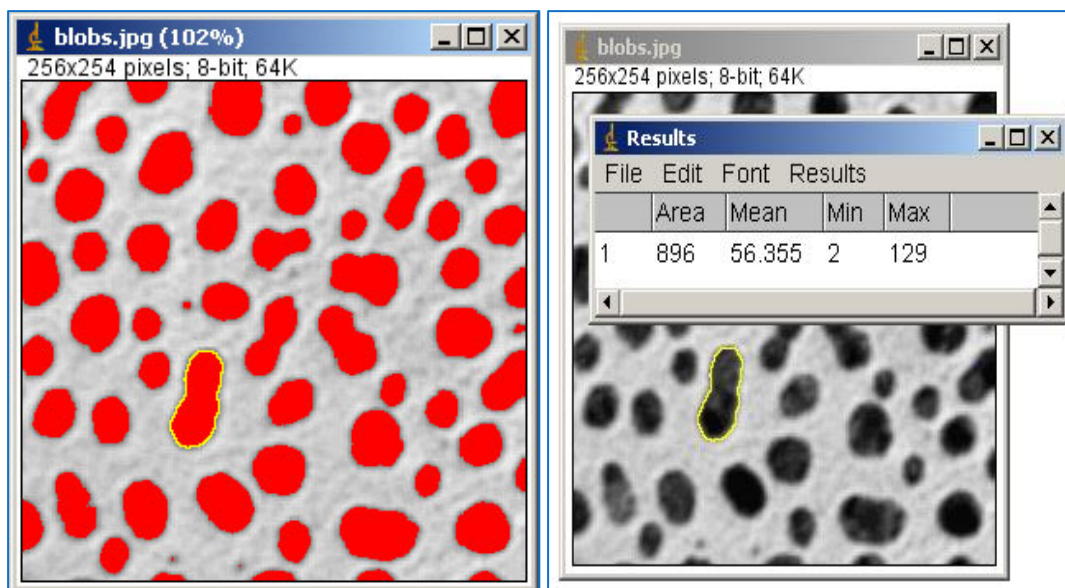


Figura 2.2.12. Izquierda: imagen umbralizada y una parte de su segmento de interés seleccionado con *Wand tracing tool*. Derecha: características de la selección (una partícula)

2.2.5. Color Threshold

Mientras que la opción anterior, *Threshold*, sólo permitía umbralizar imágenes en escala de grises, la opción *Color Threshold* proporciona las herramientas necesarias para umbralizar imágenes en color aprovechando sus propiedades:

- Tres histogramas. En los tres paneles superiores se representan los histogramas de las tres componentes de la imagen en color, sean RGB, HSB, Lab o YUV (éstas se pueden seleccionar en la lista desplegable *Color space*, o Espacio de Color, que se describe más adelante).
- Barras deslizantes de los umbrales. Al igual que en el caso del menú anterior, permiten ajustar los umbrales inferior y superior, aunque en este caso hay un par de barras por cada histograma, sobre los cuales se muestran los intervalos seleccionados con ventanas superpuestas. A la derecha de cada histograma hay una casilla con la etiqueta *Pass*. Al marcar esta casilla se considera un filtro paso banda, es decir, que los niveles de la región de interés son los acotados por los umbrales. Por el contrario, si la casilla se desmarca se considera un filtro banda eliminada y se interpreta que los niveles de la región de interés son los que quedan fuera del intervalo delimitado por los umbrales. En cualquier caso, sólo los píxeles cuyas tres componentes tengan niveles que se encuentren en los tres intervalos de paso seleccionados se consideran pertenecientes a la región de interés.
- *Thresholding method* (método de umbralización). Como ocurría en el menú *Threshold*, existe la opción de elegir el algoritmo que se emplea en caso de hacer el umbralizado de forma automática.
- *Threshold Color*. De nuevo al igual que en el caso anterior, es posible elegir cómo representar la imagen una vez umbralizada. Las opciones *Red*, *White* y *Black* hacen que la parte de interés se coloree de rojo, blanco o negro respectivamente. La opción *B&W* funciona tal y como se describió en el menú anterior.

- *Color Space* (espacio de color). Permite elegir el espacio de color en el que se va a trabajar para llevar a cabo la umbralización. Es posible elegir entre RGB y HSB, que ya se han comentado, pero también están disponibles los espacios de color Lab e YUV:
 - Lab. En realidad bajo esta denominación se agrupan dos espacios de color distintos, CIELAB (CIE² 1976 $L^*a^*b^*$) y Hunter Lab (Hunter L, a, b), que a pesar de ser diferentes comparten el mismo origen y la misma filosofía. Derivan del espacio maestro CIE 1931 XYZ y están orientados a ser espacios de color visualmente lineales, es decir, se pretende que exista una relación lineal entre los cambios originados en el espacio de color y los cambios percibidos por del sistema visual humano. Son modelos tridimensionales con las siguientes componentes: L^* (la luminosidad del color, que vale 0 para el negro y 100 para el blanco), a^* (la relación del color con el rojo y el verde, siendo los valores positivos los que lo acercan al primero y los valores negativos los que lo acercan al segundo) y b^* (la relación del color con el amarillo y el azul, siendo los valores positivos los que lo acercan al primero y los valores negativos los que lo acercan al segundo) [4]. ImageJ permite aprovechar estas componentes para umbralizar imágenes con color.
 - YUV. Modelo de color utilizado en televisión analógica PAL (*Phase Alternating Line* o Línea de Fase Alternada) y NTSC (*National Television System Committee* o Comisión Nacional de Sistema de Televisión). Como el resto de los espacios de color de la familia YCbCr, está orientado a separar las imágenes en una componente de brillo (luminancia, Y) y dos componentes relacionadas con el color (crominancia, U y V). Con esto se pretende principalmente aumentar la eficiencia de las transmisiones de vídeo e imágenes y su almacenamiento a base de reducir el ancho de banda dedicado al color de las imágenes, aprovechando las deficiencias del sistema visual humano en cuanto a la sensibilidad para el color (posibilita introducir errores sin que repercutan en la visualización). ImageJ permite utilizar las componentes Y (brillo), U (diferencia de azul) y V (diferencia de rojo) para umbralizar imágenes en color.
- *Dark background*. Funciona igual que en el apartado anterior: intercambia la forma de representar las partes de interés de la imagen por la forma de representar el fondo.
- *Original*. Devuelve la imagen a su estado original pero conserva la configuración hecha hasta el momento.
- *Filtered*. Si se ha pulsado el botón *Original* y se quiere volver a ver el resultado de filtrar la imagen con la configuración actual, este botón permite hacerlo.
- *Select*. Hace una selección automática de todos los segmentos de interés extraídos con la configuración actual.

² *Commission Internationale d'Eclairage* o Comisión Internacional de la Iluminación

- *Sample*. Permite definir los umbrales que delimitan los intervalos de interés de cada componente de forma automática en función de las componentes de los píxeles contenidos en una selección hecha previamente sobre la imagen activa.
- *Stack*. Si se trabaja con pilas de imágenes de tipo *RGB Color*, permite procesar todas las imágenes de la pila con los parámetros actuales.
- *Macro*. Crea un *macro* con la configuración actual y la manda al grabador de macros (ver el apartado 5.1 Submenú *Macros*).
- Botón *Help*. Abre una ventana con información sobre *Color Threshold* como recogida en este apartado.

Como se ha podido comprobar, con esta herramienta es posible segmentar en la imagen activa sólo aquellas partes que tienen un determinado tono, saturación y brillo, o unas determinadas componentes RGB. En la Figura 2.2.13 se muestra un ejemplo de esto.

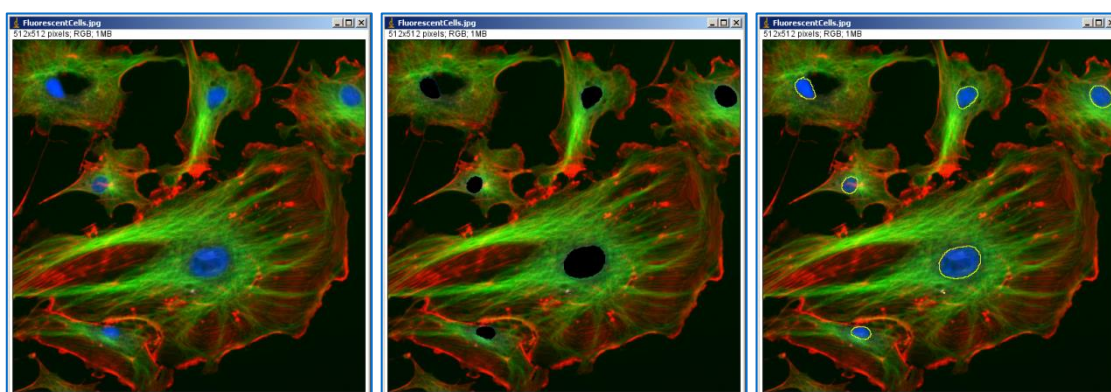


Figura 2.2.13. Izquierda: imagen original, células fluorescentes. Centro: imagen umbralizada con las regiones de interés (los núcleos de las células) destacadas en negro. Derecha: imagen original con una selección de los núcleos hecha a partir de la umbralización previa.

A la izquierda se puede ver la imagen original, unas células fluorescentes con núcleos azules que se quieren segmentar. Se ha configurado el menú de *Color Threshold* para trabajar en el espacio de color HSB. Se han definido los umbrales de tono para que las regiones a segmentar sean aquellas con tonos azules (y ligeramente cian y magenta). Se han definido filtros paso todo para la saturación y el brillo (el tono es suficiente para discriminar los núcleos del resto de la célula). Después se ha elegido el color negro para las zonas segmentadas. De este modo, todos los píxeles que en la imagen original tuvieran una componente *H* con un valor comprendido en el intervalo delimitado por los umbrales pasan a formar parte de la región de interés segmentada y se destacan en negro. Una vez umbralizada la imagen, se ha pulsado el botón *Select* para hacer una selección automática en la imagen original de las zonas segmentadas en la umbralización.

2.2.6. Size

Esta opción permite modificar el tamaño de la imagen activa en sus dimensiones vertical y horizontal, y también la profundidad en caso de tratarse de una pila de imágenes (es decir, modifica el número de capas de

la pila), además de la dimensión temporal en las hiperpilas (modifica el número de *frames*). Su menú contextual dispone de los siguientes parámetros:

- *Width (pixels)*. Anchura en píxeles de la nueva imagen.
- *Height (pixels)*. Altura en píxeles de la nueva imagen.
- *Depth (slices)*. Profundidad en capas (entendidas como imágenes que contiene la pila) de la nueva pila.
- *Time (frames)*. Tiempo en *frames* (cuadros) de la hiperpila.
- *Constraint aspect ratio*. Restringe la relación de aspectos. Cuando se activa, la relación entre anchura y altura se mantiene constante, y cualquier cambio en una dimensión se refleja inmediatamente de forma proporcional en la otra.
- *Average when downsizing*. Marcando esta casilla, al reducir las dimensiones de las imágenes se lleva a cabo de forma previa un filtrado paso bajo (de media, de ahí el nombre *Average*) para prevenir la aparición de *aliasing* que se puede derivar de la reducción de tamaño, que al fin y al cabo es un submuestreo.
- *Interpolation*. Esta lista desplegable permite elegir el algoritmo en base al cual se calculan, interpolando, los valores de los nuevos píxeles que aparecen cuando la imagen se hace más grande. Hay dos disponibles:
 - *Bilinear* (bilineal). Con este tipo de interpolación, a la hora de calcular el valor de los nuevos píxeles se tienen en cuenta los valores de los píxeles adyacentes tanto horizontalmente como verticalmente. De hecho, se llama bilineal porque equivale a interpolar primero en horizontal y luego en vertical. Esto se pone de manifiesto en el siguiente ejemplo. En la Figura 2.2.14, a la izquierda, se muestran ordenados los niveles de brillo de los 64 píxeles de la esquina superior izquierda de una imagen de tipo *8-bit* (esto es posible gracias a las herramientas de selección y a la herramienta *Image to Results* del submenú *Transform* en el menú *Image*). A la derecha se muestran los valores de los píxeles una vez se ha cambiado el tamaño de la imagen al doble en ambas dimensiones. Este cambio conlleva la aparición de nuevos píxeles entre los píxeles originales cuyo valor es necesario inferir a través de la interpolación. Como se doblan dos dimensiones hay cuatro veces más píxeles, y por lo tanto aparecen tres nuevos píxeles por cada píxel original: uno a la derecha/izquierda, otro arriba/abajo y otro en diagonal. Esto se aprecia fácilmente en la figura observando los valores destacados entre la línea discontinua y su nueva posición una vez hecho el cambio de dimensiones.

	X0	X1	X2	X3	X4	X5	X6
Y0	6	6	6	6	6	6	6
Y1	71	113	49	19	33	81	93
Y2	71	91	49	60	33	60	60
Y3	27	60	33	60	49	50	33
Y4	50	33	101	101	19	49	33
Y5	71	60	71	113	14	33	19
Y6	50	33	19	50	71	49	19
Y7	93	49	19	33	60	81	50

	X0	X1	X2	X3	X4	X5	X6	X7
Y0	6	6	6	6	6	6	6	6
Y1	6	6	6	6	6	6	6	6
Y2	39	39	49	60	44	28	20	1
Y3	71	71	92	113	81	49	34	1
Y4	71	71	87	102	76	49	44	4
Y5	71	71	81	91	70	49	55	6
Y6	49	49	62	76	58	41	51	6
Y7	27	27	44	60	47	33	47	6

Figura 2.2.14. Izquierda: niveles de brillo de los 64 píxeles de la esquina superior izquierda de una imagen de tipo 8-bit. Derecha: niveles de brillo de los 64 píxeles de la esquina superior izquierda de la misma imagen con unas dimensiones que son el doble de las originales

La interpolación lineal se implementa como sigue:

$$y = \frac{(x - x_1)}{(x_2 - x_1)}(y_2 - y_1) + y_1 \quad (2.2.13)$$

donde, en este caso, x es la coordenada horizontal o vertical del píxel, y es el nivel de brillo del píxel, x_1 y x_2 son las coordenadas de los píxeles vecinos e y_1 y y_2 son los niveles de brillo de los píxeles vecinos. Teniéndolo en cuenta, una primera interpolación horizontal sobre los píxeles destacados en la figura resultaría en lo siguiente (en los bordes de la figura aparecen las coordenadas de los píxeles):

$$y = \frac{(4 - 3)}{(5 - 3)}(113 - 49) + 49 = 81$$

$$y = \frac{(4 - 3)}{(5 - 3)}(91 - 49) + 49 = 70 \quad (2.2.14)$$

Es decir, tras la primera interpolación lineal, el resultado sería el siguiente:

Tabla 2.2.1. Primera interpolación lineal de una interpolación bilineal

X3	X4	X5
113	81	49
91	70	49

Para terminar de realizar la interpolación bilineal es necesario hacer de nuevo una interpolación lineal en sentido vertical sobre el resultado de la primera interpolación lineal:

$$\begin{aligned}
 y &= \frac{(4-3)}{(5-3)}(113-91) + 91 = 102 \\
 y &= \frac{(4-3)}{(5-3)}(81-70) + 70 \approx 76 \\
 y &= \frac{(4-3)}{(5-3)}(49-49) + 49 = 49
 \end{aligned}
 \tag{2.2.15}$$

Así que el resultado final de la interpolación bilineal sería:

Tabla 2.2.2. Segunda interpolación lineal de una interpolación bilineal

	X3	X4	X5
Y3	113	81	49
Y4	102	76	49
Y5	91	70	49

Como se puede observar, el contenido de la Tabla 2.2.2 coincide con la Figura 2.2.14.

- *Bicubic* (bicúbica). Este tipo de interpolación tiene en cuenta más información a la hora de inferir los niveles de brillo de los nuevos píxeles (lo que resulta en un interpolado más suave y de mayor calidad), aunque por otro lado conlleva una mayor carga computacional y es más lenta que la interpolación bilineal. Si en el primer caso se consideraban 4 píxeles adyacentes (2 por cada dimensión), con esta interpolación se toman en cuenta 16. Por lo demás, la filosofía que se sigue es la misma.

2.2.7. Canvas size

Este menú permite cambiar las dimensiones del lienzo sobre el que se muestra la imagen activa. Se da la opción de especificar en píxeles la anchura (*Width*) y la altura (*Height*) del lienzo. Si las nuevas dimensiones del lienzo son más grandes que las dimensiones de la imagen, se añade un fondo blanco (y si son demasiado grandes para la pantalla, se muestra una versión reducida del lienzo conteniendo la imagen). Si las nuevas dimensiones son más pequeñas que las de la imagen, se muestra una versión recortada de la imagen. Con la lista desplegable *Position* se puede especificar en qué posición relativa al lienzo queda la imagen si se agrandan las dimensiones del lienzo y se muestra el fondo blanco; en caso de que las dimensiones del lienzo sean más pequeñas que las de la imagen, esta lista desplegable sirve para especificar qué parte de la imagen se recorta. Por último, la casilla *Zero Fill* ofrece la opción de rellenar el fondo del lienzo de negro en vez de blanco.

2.2.8. Line Width

Al seleccionar esta opción se despliega un menú contextual que permite definir el grosor de la línea de las selecciones lineales.

Figura 2.2.15. Menú *Line Width*

Es posible definirlo gráficamente con una barra deslizante o de forma numérica en un cuadro de texto. Además aparece la opción *Spline Fit*. Si se usa una selección lineal de línea segmentada, al marcar esta opción se acomoda un *spline* (curva diferenciable definida en porciones mediante polinomios) a la línea segmentada:

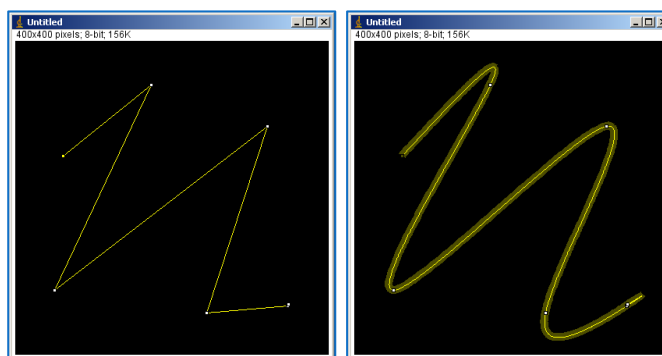


Figura 2.2.16. Izquierda: línea de selección segmentada. Derecha: línea de selección creada a partir de un *spline* acomodado a la línea anterior (y con mayor grosor, 10).

2.3. Show Info

Esta opción despliega una ventana de texto en la que se muestra información sobre la imagen activa. Esta información se puede copiar al portapapeles del sistema o guardar en un fichero de texto (haciendo *click* derecho sobre el texto).

- *Title*. Título
- *Width*. Anchura en píxeles
- *Height*. Altura en píxeles
- *Depth*. Para pilas e hiperpilas, número de capas. Este campo no aparece cuando se trata de imágenes individuales.
- *Pixel/Voxel size*. Tamaño del píxel (o vóxel). Este parámetro no está directamente relacionado con la información visual de la imagen, sino con los valores de las coordenadas de las píxeles. Por defecto vale 1x1, lo que quiere decir que el primer píxel que hay en la bisectriz del primer cuadrante después del origen de coordenadas tiene coordenadas 1,1. Si el tamaño del píxel cambiara a $m \times n$, el mismo píxel tendría coordenadas m, n , y el siguiente, $2m, 2n$, pero la información visual no se modificaría. Este parámetro se puede modificar eligiendo la opción *Properties* del menú *Image* y cambiando los valores de *Pixel Width* y *Pixel Height* (pueden incluso tomar valores negativos, lo cual repercute en coordenadas negativas). Si en lugar de imágenes individuales se trata de pilas o hiperpilas, en vez del tamaño del píxel

se describe el tamaño del vóxel, que es un píxel con coordenada de profundidad. Por defecto el tamaño de vóxel es $1 \times 1 \times 1$, así que el píxel que sigue en diagonal al origen de coordenadas de la capa n tendría coordenadas $1, 1, n$. Si el voxel tuviera tamaño $1 \times 1 \times 2$, el mismo píxel tendría coordenadas $1, 1, 2n$. También se puede modificar cambiando el valor de *Voxel Depth* en la opción *Properties* del menú *Image*.

- *ID*. Número negativo que sirve de identificador único para cada imagen que se abre en una sesión de ImageJ (una misma imagen abierta dos veces en dos ventanas diferentes supone dos identificadores)
- *Coordinate origin*. Distancia del origen de coordenadas a la esquina superior izquierda de la imagen. Se trata de otro parámetro que no está directamente relacionado con la información visual de la imagen, sino con los valores de las coordenadas de las píxeles. Por defecto vale 0×0 , lo que implica que el origen de coordenadas de la imagen está en su esquina superior izquierda. Se puede modificar eligiendo la opción *Properties* del menú *Image*, y es posible indicar valores tanto negativos (lo que supondría que la esquina superior izquierda tendría coordenadas m, n positivas) como positivos (lo que supondría que la esquina superior izquierda tendría coordenadas m, n negativas).
- *Bits per pixel*. Bits necesarios para describir un píxel (n para las imágenes de tipo *n-bit*, 8 para las imágenes de tipo *8-bit Color*, 32 para las imágenes de tipo *RGB-Color*,...).
- *Display range*. Para imágenes en escala de grises, el rango de niveles de gris que se muestra (siempre 0-255 para imágenes tipo *8-bit* y otros rangos para imágenes de tipos con mayor profundidad de bit). Para otro tipo de imágenes, este campo no aparece.
- *Image n/m*. Cuando se trata de pilas o hiperpilas, número de la capa que se visibiliza (n) y el total de capas de la pila (m).
- *Threshold/No Threshold*. Indica si la imagen ha sido o no umbralizada, y en caso de que haya sido umbraliza muestra el margen dinámico umbralizado.
- *Calibrated/Uncalibrated*. Indica si la imagen está o no calibrada en densidad (ver el apartado 4.9 *Calibrate*).
- *Path*. Ruta del archivo que contiene la imagen
- *Selection/No selection*. Indica si hay o no una selección sobre la imagen, y en caso de haberla se describe (tipo de selección, anchura, altura,...).
- Otros. Por ejemplo, para imágenes en formato jpg se indican parámetros jpg, como el tipo de compresión, las tablas de cuantización,...

2.4. Properties

Con esta opción se posibilita la modificación directa de algunos de los parámetros de la imagen estudiados en el menú anterior.

- *Channels, Slices y Frames*. Permite modificar el número de canales, tiras y cuadros que hay en una pila o hiperpila, aunque se debe cumplir que se mantenga el número de capas, o lo que es lo mismo, el producto *Channels x Slices x Frames* debe ser igual a una constante que se indica bajo los cuadros de texto en cada caso (es decir, permite redistribuir los elementos de la pila). Para imágenes individuales *Channels, Slices y Frames* valen 1 y su valor no se puede cambiar (el producto *Channels x Slices x Frames* debe ser igual a 1).
- *Unit of Length*. Unidad de medida a la que se refieren los campos *Pixel Width*, *Pixel Height* y *Voxel Depth*. Es una cadena de caracteres que puede valer “pixels” o, en caso de que la imagen esté calibrada espacialmente, “nm”, “μm”,...No influye en la información visual de la imagen, tan sólo en las unidades de los parámetros espaciales de la imagen.
- *Pixel Width y Height, y Voxel Depth*. Como se comentó en el apartado anterior, permiten especificar las dimensiones de los píxeles de las imágenes sencillas (o de los vóxeles en el caso de las pilas o hiperpilas). No afecta a la información visual de las imágenes, tan sólo a los valores de las coordenadas de sus píxeles.
- *Frame Interval*. En el caso de las hiperpilas con dimensión temporal (que se pueden animar), en este campo se puede especificar el intervalo de tiempo que transcurre entre dos cuadros consecutivos. El valor mínimo que se puede especificar es 0, que es el valor que aparece por defecto. Cuando aparece este valor, el intervalo de tiempo entre cuadro y cuadro es el que se hay especificado en el campo *Speed* de las opciones *Animation Options* del apartado *Tools* del submenú *Stacks*, menú *Image* (realmente es el inverso del valor que aparece en dicho campo, puesto que especifica en fps³).
- *Origin (pixels)*. Permite establecer la situación del origen de coordenadas en la imagen. Como ya se comentó, corresponde a la distancia del nuevo origen de coordenadas a la esquina superior izquierda de la imagen (por defecto vale 0,0, así que el origen de coordenadas se encuentra en su esquina superior izquierda). Se trata de otro parámetro que no está directamente relacionado con la información visual de la imagen, sino con los valores de las coordenadas de las píxeles.
- *Global*. Marcando esta opción, los cambios hechos en el menú emergente de propiedades tendrán carácter global y afectarán a todas las imágenes que se abran durante la sesión de ImageJ. En caso contrario, afectarán sólo a la imagen activa.

2.5. Submenú Color

En este submenú se encuentran las herramientas que tratan las características exclusivas de las imágenes en color. Si se intenta utilizarlas con imágenes en escala de grises, generalmente aparece el siguiente error:

³ *Frames Per Second* o Cuadros Por Segundo

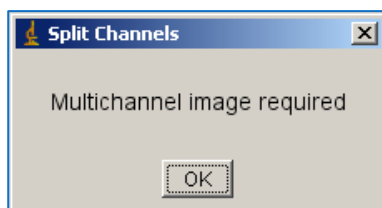


Figura 2.5.1. Ventana de error que aparece cuando se pretende usar las herramientas del menú *Color* sobre imágenes en escala de gris. Contiene el mensaje *Multichannel image required* (Imagen multicanal requerida).

2.5.1. Split Channels

Con una imagen individual de cualquier tipo de imagen en color, esta opción (que significa Separar Canales) elimina la imagen en color y en su lugar genera tres imágenes en escala de grises (de tipo *8-bit*) que corresponden a las tres componentes RGB de la imagen original, de modo que el nivel de gris de los píxeles de estas nuevas imágenes corresponde a la intensidad de rojo, verde y azul de los píxeles de la imagen original. Por ejemplo, en la imagen de la izquierda de la Figura 2.5.2, que corresponde al plano rojo de la imagen baboon.bmp, el morro del babuino tiene un gran nivel de gris (es prácticamente blanco) porque en la imagen original el morro tenía una gran cantidad de rojo. Las imágenes aparecen en orden y etiquetadas en su parte superior con las palabras *red*, *green* y *blue* según corresponda.

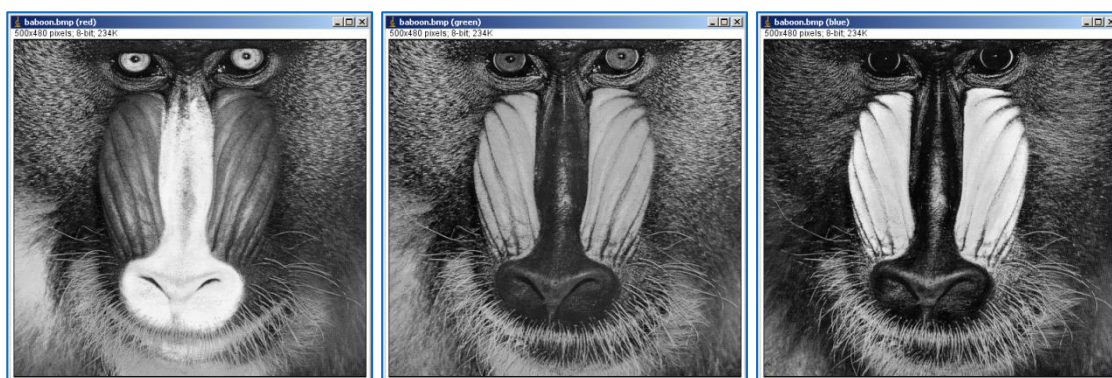


Figura 2.5.2. Resultado de aplicar la herramienta *Split Channels* sobre la imagen baboon.bmp: de izquierda a derecha, imágenes de tipo *8-bit* correspondientes a sus capas roja, verde y azul.

En el caso de pilas de imágenes RGB, esta opción resulta en la eliminación de la pila original y la generación de tres nuevas pilas, en cada una de las cuales aparecen agrupadas las componentes de las imágenes que había en la pila original (una pila para las componentes rojas, otra para las componentes verdes y otra para las componentes azules).

Por último, si se usa con imágenes compuestas, en las que los colores se manejan a través de canales separados, o con hiperpilas en la que una dimensión son los canales, resulta en la división de la imagen compuesta o de la hiperpila en los canales, que no se muestran agrupados, sino aislados en ventanas diferentes.

2.5.2. Merge Channels

Es la herramienta inversa a la anterior, *Split Channels*, literalmente significa Fundir Canales. Sin embargo, como el proceso de fundir no es tan sencillo como el de separar, ofrece algunas opciones más, que se presentan al usuario en un menú emergente que contiene los siguientes elementos:

- Listas desplegables C1 – C7. Permiten especificar qué imágenes han de tomarse para cada componente RGB de la imagen que resultará de hacer el fundido. La única restricción es que las imágenes que se indiquen no pueden tener dimensiones diferentes (en ese caso se muestra un mensaje de error). Por lo demás no hay limitaciones, así que podrían intercambiarse de orden las componentes de una misma imagen o incluso tomarse componentes de imágenes diferentes que tengan el mismo tamaño, lo cual resulta en imágenes como la de la Figura 2.5.3. Ésta muestra una mezcla en la que la componente roja corresponde al plano rojo de lena.jpg, la componente verde corresponde al plano verde de baboon.jpg y la componente azul corresponde al plano azul de lena.jpg. Por ello se observa la fusión del rostro del babuino exclusivamente con tonos verdes y el rostro de Lena sin tonos verdes.

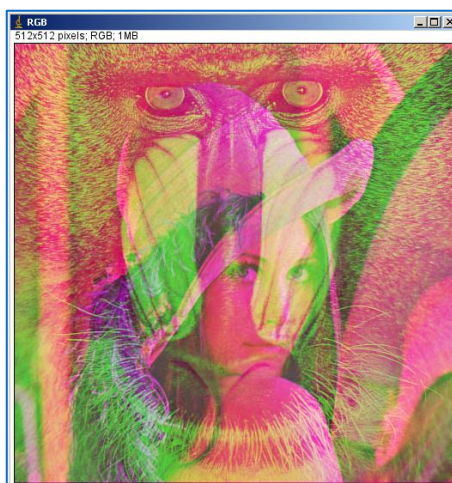


Figura 2.5.3. Imagen resultado de hacer una fusión en la que las componentes roja y azul corresponden a los planos rojo y azul de lena.jpg, mientras que la componente verde corresponde al plano verde de baboon.jpg

También existe la posibilidad de especificar otras componentes de la imagen, aunque en caso de hacerlo no resulta en una imagen de tipo *RGB Color*, sino en una imagen compuesta (esté o no marcada la casilla *Create composite*, que se aborda más adelante). Se puede especificar qué imagen se toma para las componentes gris (*grey*), cian (*cyan*), magenta (*magenta*) y amarilla (*yellow*). A pesar de que ya se ha tratado la relación entre las componentes RGB y la escala de grises de una imagen (ver el apartado 2.1 Submenú *Type*), no se ha tratado la relación de las componentes RGB con las componentes CMY (*Cyan*, *Magenta*, *Yellow*). Mientras que las componentes RGB se enmarcan en un modelo de color aditivo y se refieren a las cantidades de rojo, verde y azul que hay que emitir para conseguir determinado color en base a las leyes de la colorimetría tricromática [5], las componentes CMY se utilizan en mezclas

sustractivas e indican qué zonas del espectro de la luz blanca (en principio) hay que absorber para conseguir un color específico a base de mezclar tinturas cian, magenta y amarilla. La tintura cian absorbe la parte roja de la luz e iluminada con luz blanca devuelve el color que conocemos como cian, mezcla de azul y verde; la tintura magenta absorbe la parte verde de la luz e iluminada con luz blanca devuelve el color que conocemos como magenta, mezcla de rojo y azul; y por último, la tintura amarilla absorbe la parte azul de la luz e iluminada con luz blanca devuelve el color que conocemos como amarillo, mezcla de verde y rojo. Por todo ello, las componentes RGB y CMY se relacionan así (supuesto que ambas se describen con 256 niveles de intensidad, de 0 a 255):

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5.16)$$

En la Figura 2.5.4 se muestran las componentes CMY de la imagen lena.jpg representadas como imágenes en escala de grises en las que el nivel de gris representa la cantidad de cian, magenta y amarillo que hay que mezclar para recuperar la imagen original:

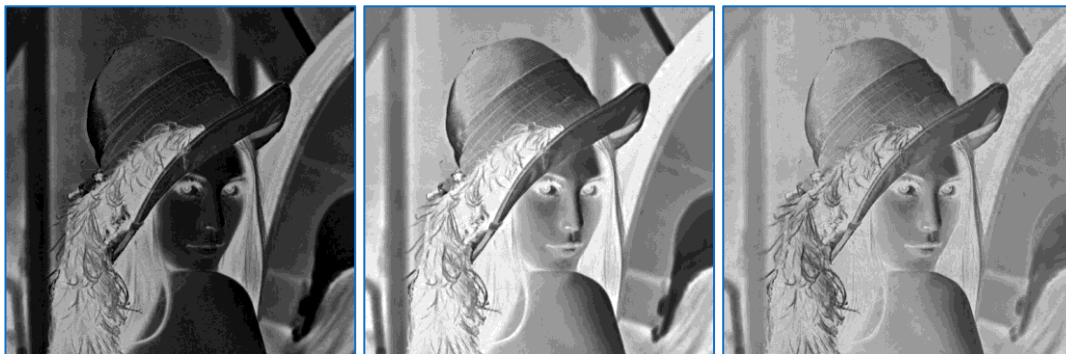


Figura 2.5.4. Componentes CMY de lena.jpg; de izquierda a derecha, componente cian, componente magenta y componente amarilla

- *Create composite.* Trabajando sólo con las componentes RGB, si se marca esta casilla se crea una imagen compuesta multicanal en vez de la imagen de tipo *RGB Color*. La imagen compuesta se crea automáticamente siempre que se tenga en cuenta una componente que no sea RGB.
- *Keep source images.* Si se marca esta casilla, se conservan las imágenes que se funden para crear la imagen final. En caso contrario, se eliminan una vez se ha creado la imagen definitiva.
- *Ignore source LUTs.* Cuando se crean imágenes de tipo *RGB Color* siempre se ignoran las tablas LUT de las imágenes que se van a fundir, pero en caso de crear imágenes compuestas, las tablas LUT se conservan a no ser que se marque esta casilla.

2.5.3. Arrange Channels

Esta herramienta permite reordenar los canales de una imagen compuesta multicanal. Para ello ofrece un menú como el que se muestra en la Figura 2.5.5:

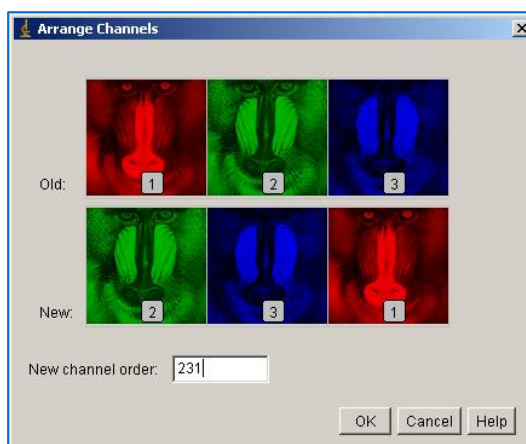


Figura 2.5.5. Menú *Arrange Channels*. En este caso se están reordenando los canales de una imagen compuesta multicanal con canales R, G y B.

La operación que se muestra en la Figura 2.5.5 no afecta al color de la imagen, tan sólo al orden en el que se muestran los canales en una imagen compuesta. Por ejemplo, después de pulsar OK, el primer canal o canal 1/3 es el verde, el canal 2/3 es el azul y el canal 3/3 es el rojo (como se muestra en la Figura 2.5.6):

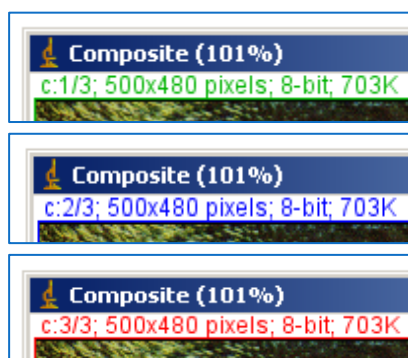


Figura 2.5.6. Canales reordenados de una imagen compuesta multicanal

2.5.4. Channels Tool

En las imágenes compuestas los colores se manejan a través de canales. Esto conlleva ciertas ventajas, como poder fundir o mantener separados más de tres canales, o activar o desactivar los canales de forma independiente (lo cual permite, por ejemplo, llevar a cabo medidas en un canal mientras se visualizan más) [6]. Esta última funcionalidad se administra a través del menú emergente que se despliega al seleccionar *Channels Tool*.

El menú consta de una lista desplegable y tantas casillas como canales tenga la imagen compuesta. La lista permite determinar el modo de visualización de la imagen compuesta y ofrece las siguientes opciones:

- *Composite*. La imagen se muestra en modo compuesto y es posible ver al mismo tiempo todos los canales que se deseen. Para activar o desactivar canales sólo hay que marcar o desmarcar sus correspondientes casillas.
- *Color*. Sólo se muestra un canal a la vez, y se muestra coloreado según el color al que corresponda. Para elegir el canal que se muestra sólo hay que marcar su casilla.
- *Grayscale*. Sólo se muestra un canal a la vez, y se muestra en escala de grises (es decir, la intensidad del color se representa mediante el nivel de brillo). Para elegir el canal que se muestra sólo hay que marcar su casilla.

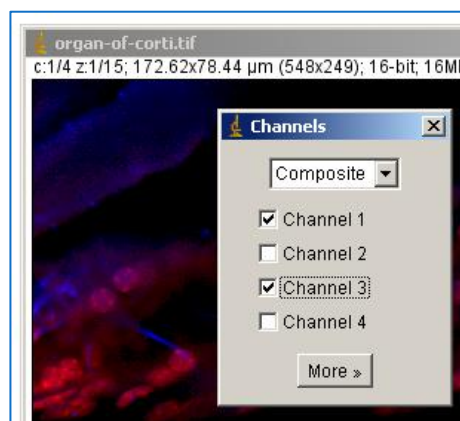


Figura 2.5.7. Imagen compuesta con los canales R y B activos

Por último, el menú contiene un botón, *More*, que despliega una lista con algunas de las opciones del submenú *Color*, que se trata en este apartado, como *Make Composite*, *Convert to RGB*, *Split Channels*,... Además, en la parte inferior de la lista que se despliega, se ofrecen varios colores para pseudocolorear la imagen compuesta (si, por ejemplo, se seleccionara *Magenta* para el canal rojo, dicho canal pasaría a contener niveles de magenta en vez de niveles de rojo).

Para terminar hay que señalar que esta opción despliega el mismo menú que se abre con la opción *Channels Tool* del submenú *Hyperstacks*, menú *Image*.

2.5.5. Stack to RGB

Esta herramienta convierte imágenes compuestas o pilas de 2 o 3 capas de imágenes de tipo *8-bit* o *16-bit* en imágenes de tipo *RGB Color* (asumiendo que están orden, es decir, que el primer plano se toma como R, el segundo como G y el tercero como B). Para realizar la conversión, directamente se consideran los niveles de color de los canales o los niveles de brillo de las capas como niveles de rojo, verde y azul.

Si se lleva a cabo sobre imágenes compuestas de sólo dos canales o pilas de sólo dos capas el plano B se considera vacío. Esto se observa en la Figura 2.5.8.

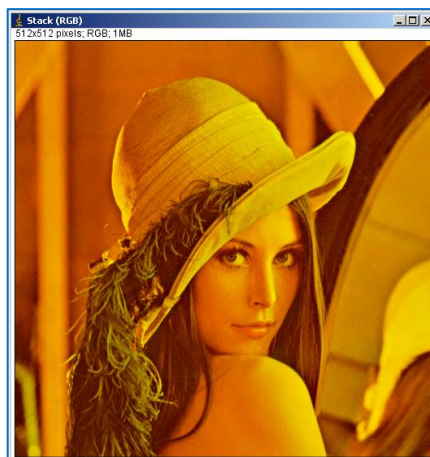


Figura 2.5.8. Imagen de tipo *RGB Color* construida con sólo dos componentes

La imagen que se muestra se ha construido aplicando la opción *Stack to RGB* sobre una pila de dos capas que contenía los planos R y G de la imagen lena.jpg. Como no se han pasado tres planos, el plano B se considera vacío y la imagen sólo contiene información del color rojo y del color verde, y de ahí su tono amarillo.

2.5.6. Make Composite

Con esta opción es posible convertir a imágenes compuestas imágenes de tipo *RGB Color*, pilas de 2 a 7 capas e hiperpilas de entre 2 y 7 canales. La conversión únicamente consiste en separar y mostrar de forma independiente los planos R, G y B implícitos en las imágenes de tipo *RGB Color*, o las capas y canales en el caso de pilas e hiperpilas. Una vez hecha la conversión se abre de forma automática el menú *Channels Tool* para administrar la imagen compuesta resultante.

2.5.7. Show LUT

Como ya se ha comentado, existen imágenes en las que los píxeles, en vez de describirse con niveles de brillo o niveles de sus componentes, se describen con índices que se refieren a las entradas de una tabla asociada a la imagen. Ésta se llama tabla LUT o *Lookup Table* (Tabla de Consulta) y es una tabla que alberga entradas que contienen la descripción de colores en base a los niveles de sus componentes RGB, que se describen cada una con 8 bits. Mediante la opción *Show LUT* es posible visualizar una representación gráfica de las LUTs de aquellas imágenes que las tengan, que básicamente son aquellas indexadas o de tipo *8-bit Color*, aunque también se puede asociar LUTs a imágenes en escala de grises. La representación gráfica consiste en un gráfico en el que el eje de abscisas contiene los índices de la tabla y el eje de ordenadas contiene los niveles de las componentes. Para cada índice hay tres puntos coloreados correspondientes a los niveles de las componentes que generan el color asociado a dicho índice, y que junto al resto de puntos forman tres curvas. Bajo el eje de abscisas se muestra una barra coloreada con los colores correspondientes a los índices. Un ejemplo se muestra en la Figura 2.5.9:

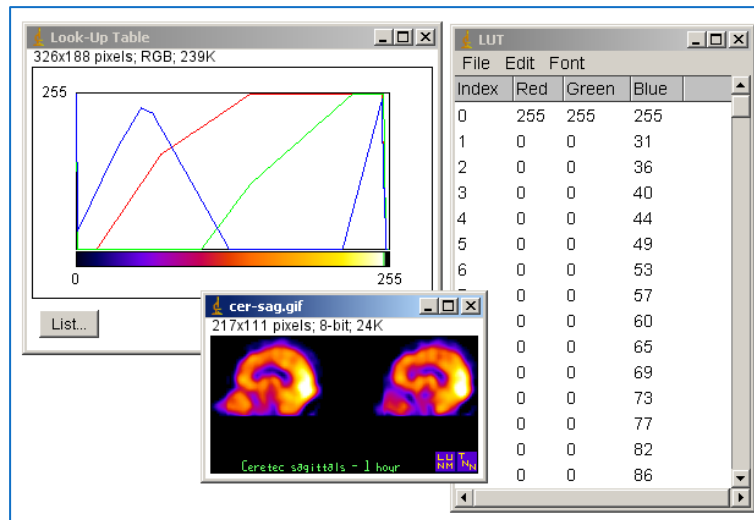


Figura 2.5.9. Imagen indexada (tipo 8-bit Color) junto a las representaciones gráfica y escrita de su LUT

El botón *List* genera la lista o representación escrita de la gráfica, la cual se puede editar y guardar. Todo ello se muestra en la Figura 2.5.9.

2.5.8. Display LUT

Al pulsar sobre *Display LUT* se muestra una imagen (recogida en la Figura 2.5.10) con unas barras coloreadas con los colores de todas las LUTs que incluye por defecto ImageJ. En el submenú *Lookup Tables*, en este mismo menú, es posible cambiar la tabla LUT de una imagen por una de estas tablas.

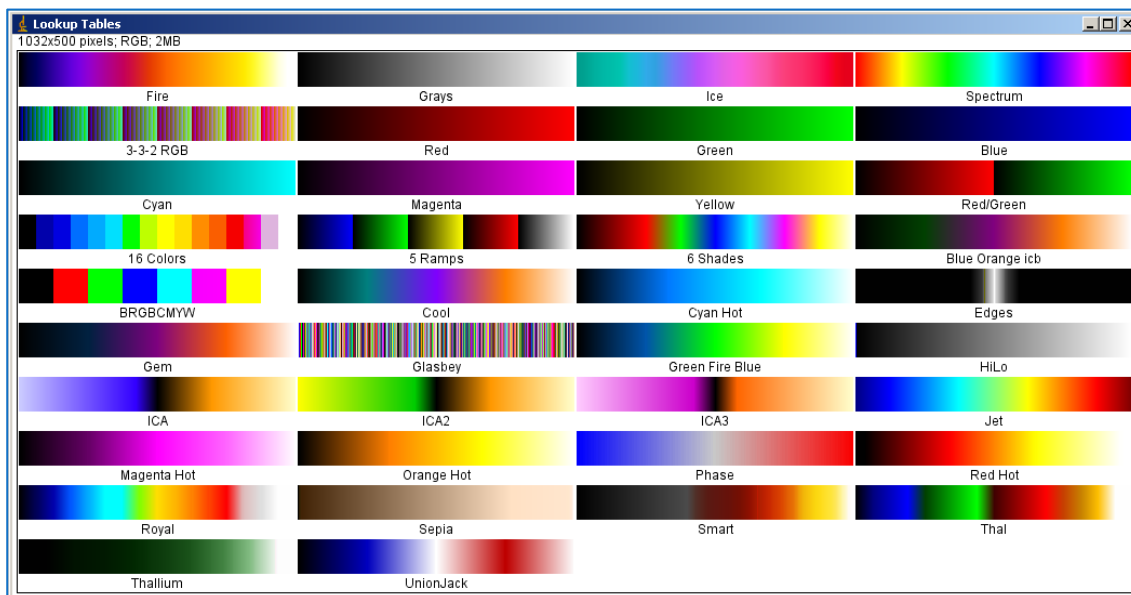


Figura 2.5.10. Tablas LUTs incluidas por defecto en ImageJ

2.5.9. Edit LUT

Abre el editor de LUTs. Éste consiste básicamente en una cuadrícula de 16x16 casillas que representan las 256 entradas de una tabla LUT (representada en la Figura 2.5.11). Al mover el cursor sobre ellas, en la barra de estado de ImageJ se muestra el índice y las componentes del color almacenado en cada entrada. Al hacer

doble *click* sobre una casilla se abre el selector de color, que permite determinar el nuevo color que contendrá la entrada asociada especificando sus componentes RGB de forma numérica o moviendo unos controles deslizantes (además muestra un cuadro con el color equivalente y su código hexadecimal).

Además, existen los siguientes botones:

- *Open*. Carga una tabla LUT en el formato propio de ImageJ (.lut), en formato *raw* o en formato de texto.
- *Save*. Permite guardar la tabla LUT actual (por defecto, como un archivo .lut).
- *Set*. Permite especificar el número de colores que puede albergar la LUT (entre 2 y 256)
- *Invert*. Invierte la tabla LUT: el índice 0 pasa a ser el 255, el 1 pasa a ser el 254, y así sucesivamente.

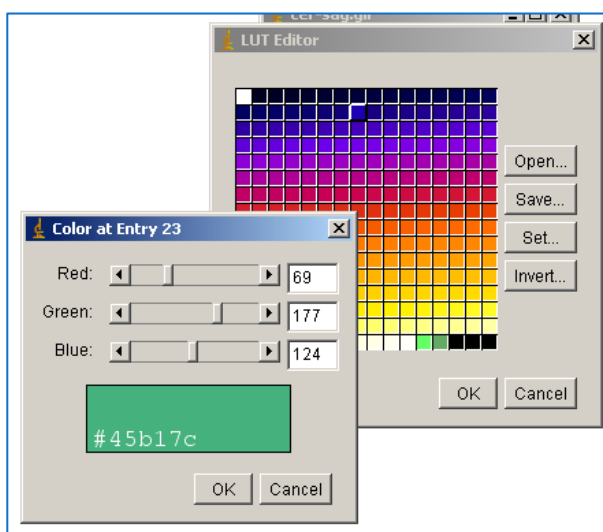


Figura 2.5.11. Editor de LUTs y selector de color

2.5.10. Color Picker

Despliega un menú que permite elegir los colores principal y de fondo que se usan con numerosas herramientas de ImageJ. Consiste principalmente en una matriz con casillas coloreadas con colores que abarcan gran parte de los tonos, saturaciones y brillos posibles, así como una barra vertical con un degradado de gris. En cualquier caso, haciendo doble *click* sobre cualquier casilla se abre el selector de color, que permite especificar 256 niveles de cada componente, por lo que es posible elegir entre 16.777.216 colores. Además, hay dos cuadros que muestran los colores elegidos, así como dos controles que permiten intercambiar entre el color principal y el color de fondo, y resetear los colores.

2.6. Submenú Stacks

En este submenú se encuentran todas las herramientas relacionadas con la gestión de pilas, las cuales se detallan a continuación.

2.6.1. Add/Delete/Next/Previous/Set Slice

Todos estos comandos posibilitan la gestión de las capas de la pila activa y por ello se encuentran agrupados dentro del submenú.

- *Add Slice.* Añade una capa vacía (negra) tras la capa que se esté visualizando en el momento en la pila activa. Si mientras se selecciona se pulsa la tecla “ALT”, la capa se añade antes de la capa que se esté visualizando. Funciona también con hiperpilas, aunque en ese caso hay que determinar mediante un sencillo menú si lo que se añade es un canal (*channel*), una capa (*z-slice*) o un cuadro (*t-frame*).
- *Delete Slice.* Elimina la capa que se esté visualizando en el momento en la pila activa. Funciona también con hiperpilas, aunque en ese caso hay que determinar mediante un sencillo menú si lo que se elimina es un canal, una capa o un cuadro.
- *Next Slice.* Muestra la capa que sigue a la que se esté visualizando en el momento en la pila activa. Para avanzar de diez en diez capas hay que pulsar las teclas “ALT” y “>” al tiempo que se selecciona.
- *Previous Slice.* Muestra la capa que precede a la que se esté visualizando en el momento en la pila activa. Para retroceder de diez en diez capas hay que pulsar las teclas “ALT” y “<” al tiempo que se selecciona.
- *Set Slice.* Muestra la capa que se especifique numéricamente mediante un menú emergente.

2.6.2. Images to Stack

Esta herramienta permite construir una pila a partir de las imágenes abiertas en la sesión de ImageJ actual, las cuales pasan a ser las capas (en el orden en que se abrieron). Evidentemente este proceso no resulta tan sencillo si las imágenes no tienen las mismas dimensiones; en tal caso aparece el menú de la Figura 2.6.1. En él se avisa de que las dimensiones difieren y se muestran las dimensiones de la mayor y de la menor de las imágenes, y debajo aparecen los siguientes elementos:

- *Method.* Consiste en una lista desplegable que permite elegir el método de construcción de la pila de entre los siguientes cuatro:
 - *Copy (center).* La pila tendrá la anchura de la más ancha de las imágenes y la altura de la más alta de las imágenes, y las imágenes se incluyen en la pila como capas sin que se modifiquen sus dimensiones y centrándolas en la pila. La parte sobrante se rellena con píxeles negros.
 - *Copy (top-left).* La pila tendrá la anchura de la más ancha de las imágenes y la altura de la más alta de las imágenes, y las imágenes se incluyen en la pila como capas sin que se modifiquen sus dimensiones y haciendo coincidir su esquina superior izquierda con la esquina superior izquierda de la pila. La parte sobrante se rellena con píxeles negros.
 - *Scale (smallest).* La pila tendrá las dimensiones de la más pequeña de las imágenes, y las imágenes pasarán a formar parte de la pila como capas escalando sus dimensiones para que coincidan con las de la pila.

- *Scale (largest)*. La pila tendrá las dimensiones de la más grande de las imágenes, y las imágenes pasarán a formar parte de la pila como capas escalando sus dimensiones para que coincidan con las de la pila.
- *Name*. Nombre de la nueva pila.
- *Title Contains*. Si se introduce una cadena de caracteres en este campo sólo se añaden a la pila aquellas imágenes cuyo título contenga dicha cadena.
- *Bicubic interpolation*. Marcando esta casilla, el escalamiento de las imágenes que se hace al seguir los métodos *Scale (smallest)* y *Scale (largest)* se hace en base a la interpolación bicúbica.
- *Use Title as Labels*. Marcando esta casilla, las etiquetas de las capas (los subtítulos que aparecen en su parte superior) pasan a ser los títulos que tenían originalmente siendo imágenes.
- *Keep Source Images*. Si se activa, una vez creada la pila se mantienen las imágenes originales. De lo contrario, se eliminan.

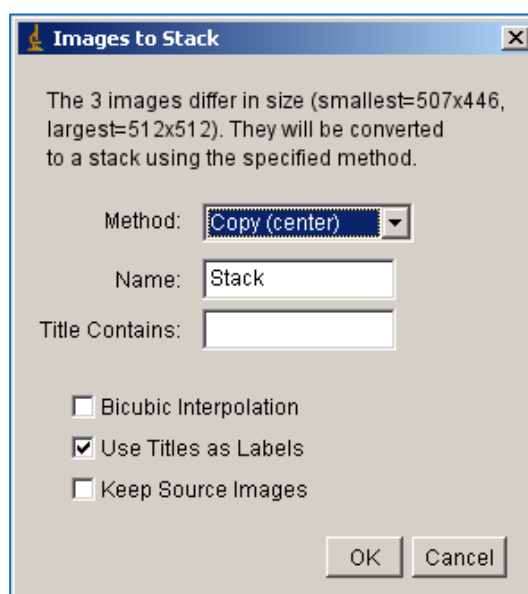


Figura 2.6.1. Menú que se despliega al intentar crear una pila a partir de imágenes de tamaños diferentes

2.6.3. Stack to Images

Esta opción deshace la pila activa y sus capas pasan a ser imágenes almacenadas en ventanas separadas.

2.6.4. Make Montage

Con esta herramienta se puede construir un montaje a partir de la pila activa. El montaje consiste en una sola imagen cuadriculada en la que se muestran todas las capas de la pila a la vez, una por cada cuadrícula, tal y como se muestra en la Figura 2.6.2:

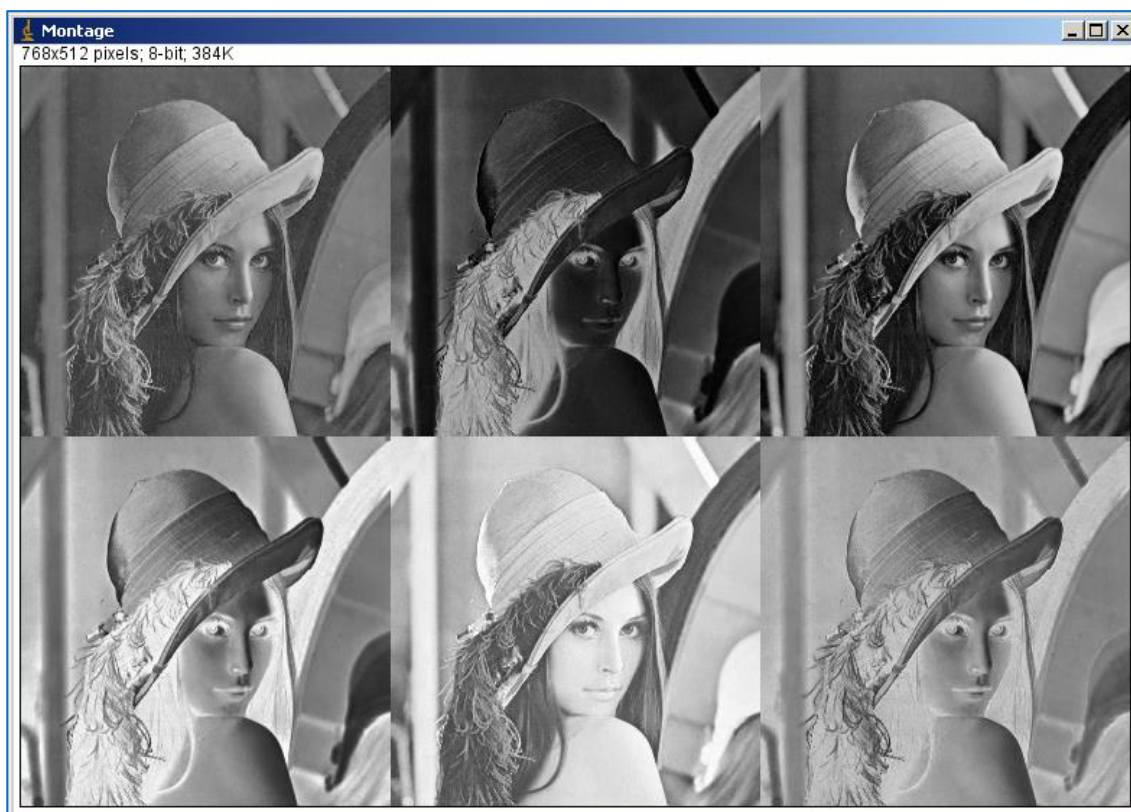


Figura 2.6.2. Montaje realizado a partir de una pila

Antes de generar el montaje aparece un menú que permite ajustar ciertos parámetros del proceso:

- *Columns y Rows.* Columnas y filas del montaje. Si el producto de ambas es mayor que el número de capas de la pila, las cuadrículas sobrantes se sitúan al final y se dejan vacías (negras). Si el producto es menor, no hay suficientes cuadrículas para albergar todas las capas y sólo se dejan las primeras capas a partir de la especificada en el campo *First Slice*.
- *Scale Factor.* Factor por el que se escalan las dimensiones de las capas al pasar al montaje.
- *First y Last Slice.* Permite especificar desde qué capa y hasta qué otra capa se introducen en el montaje. Si los valores son incongruentes (el valor de *Last Slice* es menor que el valor de *First Slice*) estos campos pasan a tomar los valores por defecto: desde la primera y hasta la última.
- *Increment.* Un incremento de valor n hace que en el montaje se incluyan las capas desde la especificada en *First Slice* y hasta la especificada en *Last Slice* cada n capas. Por defecto este campo vale 1, lo que quiere decir que se incluyen todas las capas, sin saltar ninguna.
- *Border Width.* Permite especificar el ancho del borde que se incluye entre las casillas que contienen las capas.
- *Font Size.* Permite especificar el tamaño de las etiquetas que se imprimen sobre las capas en caso de que la casilla *Label Slices* esté activada.

- *Label Slices*. Si está activada, sobre las capas colocadas en el montaje se imprimen etiquetas (que corresponden a los subtítulos que tenían en la pila).
- *Use Foreground Color*. Si esta casilla se activa, el color de los bordes y de las etiquetas pasa a ser el color principal, y el color de las casillas vacías pasa a ser el color de fondo (ver apartado 2.5.10. *Color Picker*).

2.6.5. Reslice

Para entender la función de esta herramienta hay que pensar en las pilas como volúmenes en los que las dimensiones x e y corresponden a la anchura y la altura de las imágenes, y la dimensión z corresponde a las sucesivas capas. Esto se ilustra en la Figura 2.6.3:

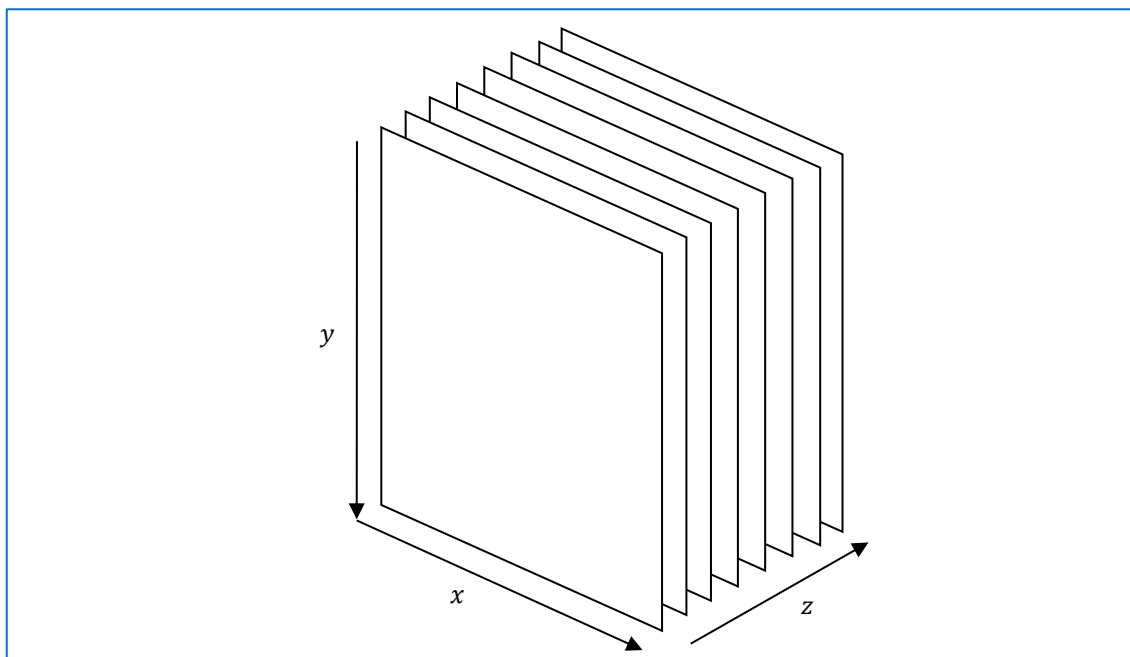


Figura 2.6.3. Ilustración del concepto de pila como volumen

Con la herramienta *Reslice* se genera a partir de las capas de la pila activa una imagen (o pila de imágenes) ortogonal a ellas. Tomando como referencia la Figura 2.6.3, la nueva imagen asociaría su anchura a la dimensión x y su altura a la z (sería un corte xz). Para indicar a qué altura y se “corta” el volumen de la pila para obtener la nueva imagen basta con hacer sobre la pila una selección lineal. También es posible hacer una selección rectangular, con lo que en lugar de una imagen ortogonal se obtienen tantas imágenes ortogonales como alturas tenga el rectángulo de la selección (y se presentan agrupadas en una nueva pila).

Esta herramienta es especialmente útil con imágenes como las contenidas en *t1-head.tiff*, producto de una resonancia magnética. Este archivo en particular contiene 129 cortes sagitales cada vez más profundos del interior de una cabeza humana. En la Figura 2.6.4 se puede observar un montaje con algunas de sus capas:

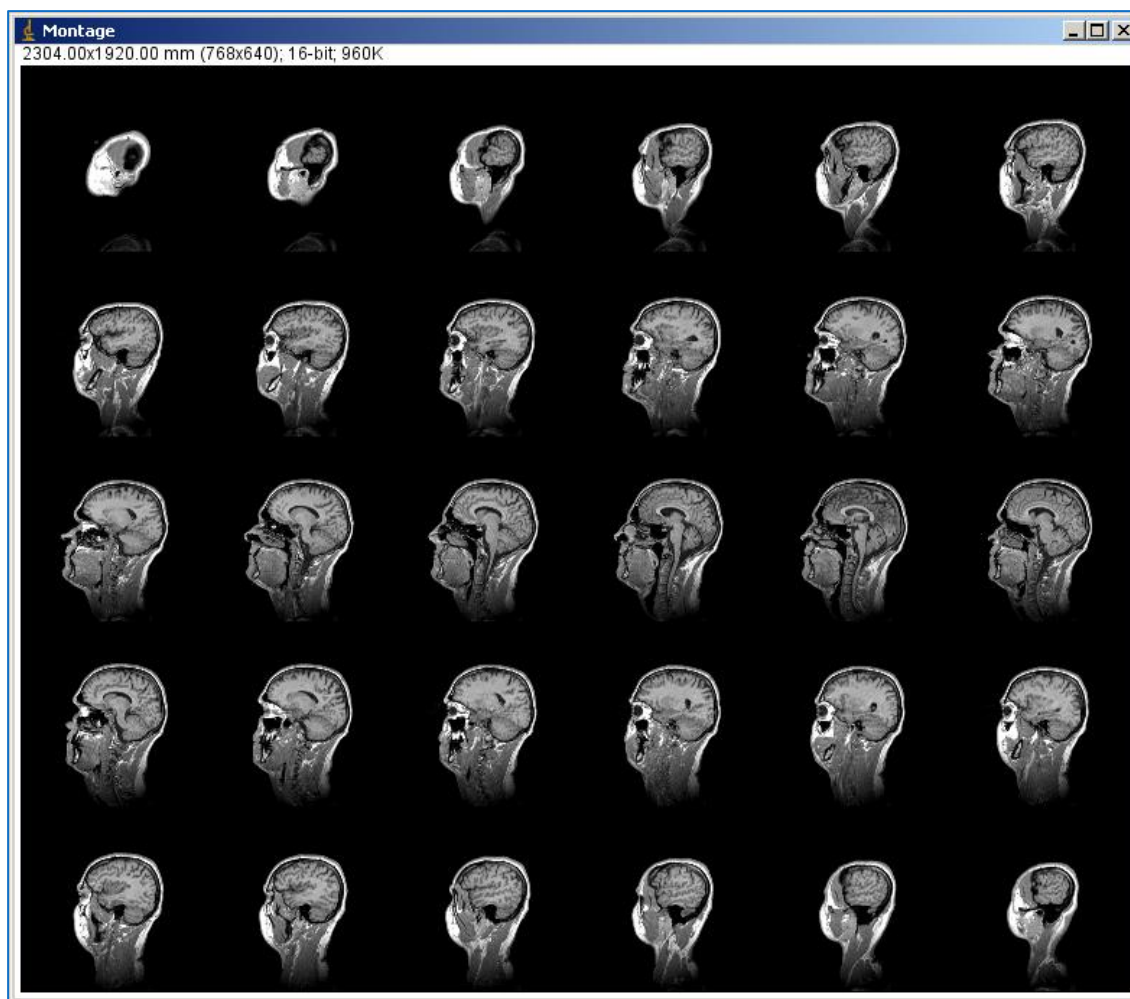


Figura 2.6.4. Montaje con las capas del archivo t1-head.tiff tomadas de 3 en 3 desde la capa 20 y hasta la capa 107

Con pilas como la de la Figura 2.6.4 es más conveniente aún si cabe su concepción como volumen, puesto que sus capas una tras otra prácticamente conforman un volumen verdadero en el que hay una cabeza humana tridimensional de la que no sólo se conoce su superficie, sino también su interior. Sin embargo, si la información se presenta como una pila sólo se pueden ver los cortes sagitales. Aquí entra en juego la herramienta *Reslice*, que permite generar cortes transversales (ortogonales a los originales) sin necesidad de repetir el proceso de captación. Esto se muestra a continuación, en la Figura 2.6.5. En la parte izquierda aparece la pila correspondiente a t1-head.tiff. La información de la pila revela que sus dimensiones son 384 mm de altura, 384 mm de anchura y 192 mm de profundidad (sus unidades son milímetros porque la pila está calibrada espacialmente; en píxeles sus dimensiones son 256x256x129). Sobre la pila hay una selección lineal de 255 mm. En estas circunstancias se emplea la herramienta *Reslice*, que resulta en la imagen de la derecha de la figura: un corte ortogonal a la pila a través de la selección lineal (a la altura de los ojos). Esta imagen es tan ancha como la selección lineal (255 mm) y tan alta como profunda es la imagen original (193,5 mm). Esto se consigue a base de muestrear cada capa en la pila a lo largo de la selección lineal de modo que el primer píxel de cada fila en la imagen de salida se toma del principio de la línea de selección en la capa

correspondiente, para después representarlo todo junto conformando la nueva imagen [7]. Al fin y al cabo, si se entiende la pila como volumen y se pasa del concepto de píxel al de vóxel, entendiéndolo como un punto del espacio con tres coordenadas y un nivel de brillo, esta operación es cualitativamente tan sencilla como reorganizar una serie de vóxeles y representarlos sobre un plano de forma que se puedan observar desde un nuevo punto de vista, pero los vóxeles siempre estuvieron ahí y seguirán siendo los mismos.

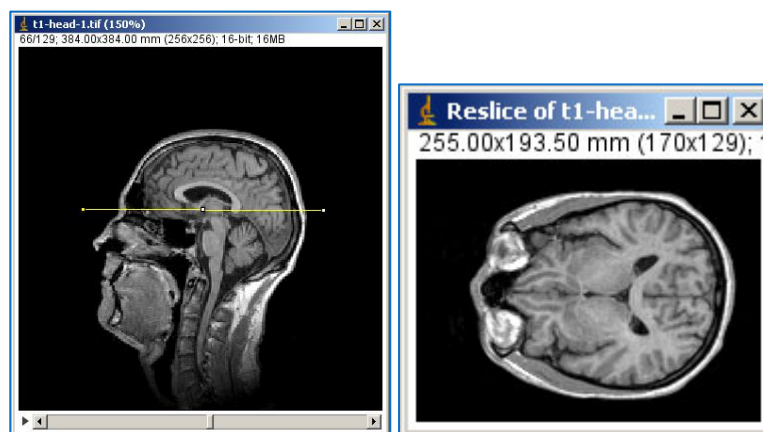


Figura 2.6.5. Izquierda: capa de profundidad 97,5 mm (de 192 mm) de la pila t1-head.tif con una selección lineal de 255 mm a la altura de los ojos. Derecha: *reslice* de la pila t1-head.tif a la altura de los ojos, de dimensiones 255 mm x 193,5 mm

Como se comentó anteriormente, esta herramienta también es aplicable con una selección rectangular, que se considerará como un conjunto de selecciones lineales tan numeroso como alta sea la selección rectangular, y por tanto da lugar a múltiples imágenes resultado, que se presentan agrupadas como capas de una pila, como se muestra en la Figura 2.6.6:

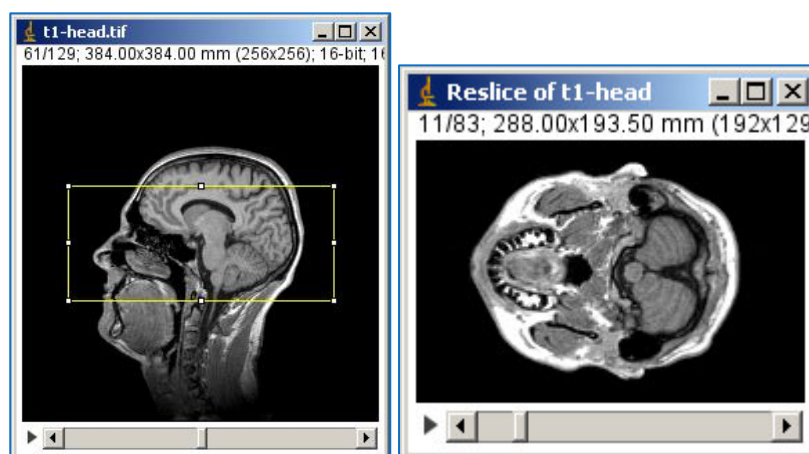


Figura 2.6.6. Izquierda: capa de profundidad 90 mm (de 192 mm) de la pila t1-head.tif con una selección rectangular de dimensiones 288x124,5 mm desde la boca hasta la frente. Derecha: Capa de profundidad 15 mm de la pila *reslice* de la pila t1-head.tif (de dimensiones 288x193,5x123 mm) a la altura de la boca

Antes de finalizar hay que comentar la existencia de un menú previo al proceso que permite modificar algunos de sus parámetros:

- *Output spacing.* Este parámetro modifica el número de capas que se tienen en cuenta para construir la imagen ortogonal (cuanto menor sea este espacio mayor información contendrá la imagen resultante y ocupará más)
- *Start at/Slice count.* Cuando la selección es rectangular, el campo *Start at* permite elegir desde dónde se empieza a “cortar” (es decir, determina qué capas aparecerán primero en la pila resultante): *top*, *left*, *bottom* y *right*, que hacen que se empiece a cortar desde arriba, desde la izquierda, desde abajo o desde la derecha respectivamente. Si la selección es lineal, este campo se llama *Slice count* y permite hacer una selección rectangular a partir de la selección lineal, de modo que la nueva selección rectangular será tan ancha como larga era la selección lineal, y tan alta como el valor que se especifique en este campo.
- *Flip vertically.* Si se marca esta casilla se realiza un *flip* vertical sobre el resultado.
- *Rotate.* Si se marca esta casilla se realiza una rotación de 90 grados sobre el resultado.
- *Avoid interpolation.* Si se marca esta casilla no se realiza ninguna interpolación, y por lo tanto el espacio tiene que ser necesariamente de 1 píxel (o su medida equivalente si la pila está calibrada), es decir, cada columna de la imagen resultante se forma tomando directamente un vóxel de cada capa.

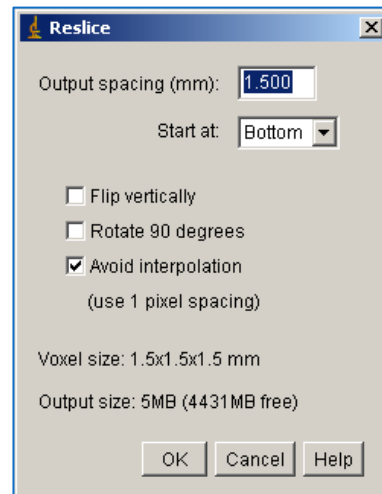


Figura 2.6.7. Menú *Reslice*

2.6.6. Orthogonal Views

Esta herramienta es similar a la anterior, aunque es interactiva, y además muestra dos vistas ortogonales: no sólo el corte *xz*, como en el caso anterior, sino también el corte *yz*. El fundamento sigue siendo el mismo. Se muestra en la Figura 2.6.8. Haciendo *click* sobre cualquier panel y arrastrando el puntero (y también cambiando de capa en el caso de la pila) cambia el contenido de los restantes paneles. La guía amarilla indica en cada panel el lugar por el que intersectan con él los planos de los paneles restantes. De esta forma se puede visualizar desde tres puntos de vista todo el contenido del volumen que supone la pila.

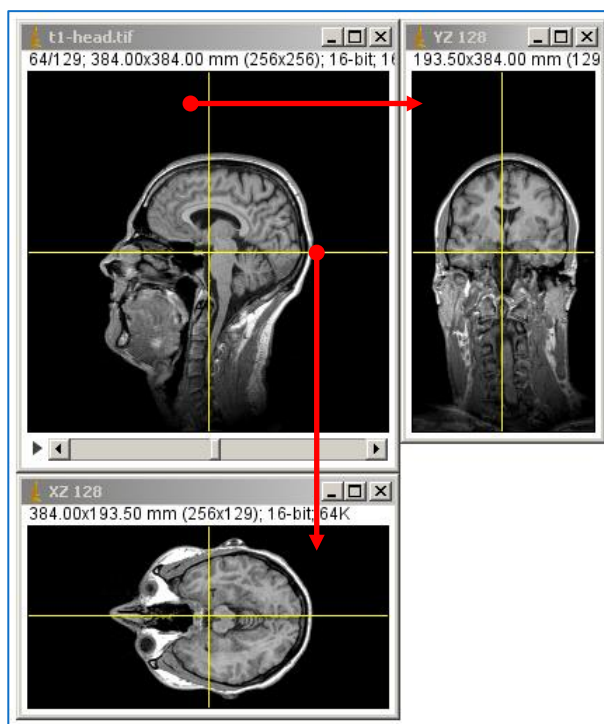


Figura 2.6.8. Herramienta *Orthogonal Views* aplicada al archivo t1-head.tif. Con las flechas rojas se destaca el hecho de que la guía amarilla indica en cada panel el lugar por el que intersectan con él los planos de los paneles restantes

2.6.7. Z Project

Como su nombre indica, esta herramienta realiza una proyección de la pila de imágenes a lo largo del eje z, de modo que el resultado es una imagen individual tan ancha y alta como ancha y alta era la pila, y en la que cada píxel tiene un valor que depende de los valores que tomaban los píxeles de idénticas coordenadas en las diferentes capas de la pila. Hay disponibles cinco modalidades de proyección que se pueden elegir en la lista desplegable del menú que aparece al seleccionar *Z Project* (que también permite determinar desde qué capa y hasta qué capa hay que considerar), y que se detallan en las siguientes líneas. Para comprobar el funcionamiento de cada una de ellas, se construye una pila con tres capas (los planos RGB de lena.jpg), y se anotan los niveles de brillo de los píxeles que tengan coordenadas $x=94$, $y=41$ (Figura 2.6.9):

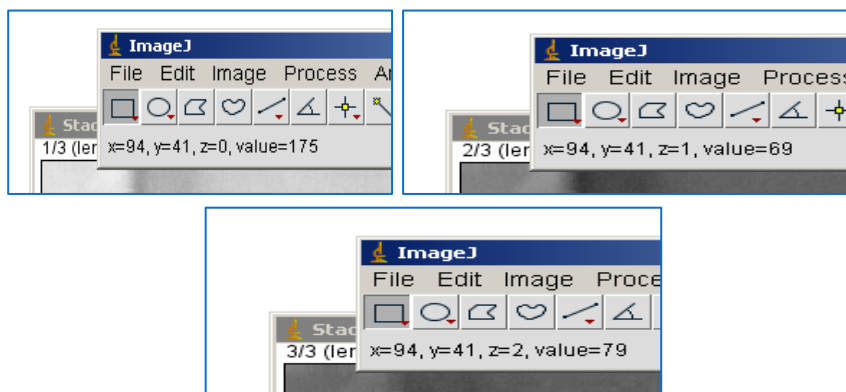


Figura 2.6.9. Niveles de brillo del píxel de coordenadas $x=94$, $y=41$: 175 en la primera capa, 69 en la segunda capa y 79 en la tercera capa.

- *Average Intensity*. Con esta modalidad de proyección, cada píxel de la imagen de salida tiene un nivel de brillo que es la media aritmética de los niveles de brillo de los píxeles que tenían sus mismas coordenadas en las capas de la pila. Se emplea con la pila de prueba y el resultado es el siguiente:

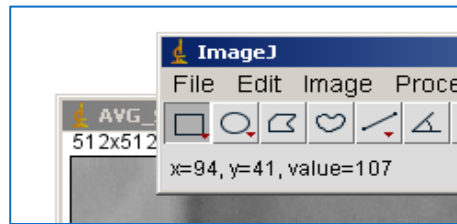


Figura 2.6.10. Nivel de brillo del píxel de coordenadas x=94 e y=41 en la proyección Z de tipo *Average Intensity* de la pila de prueba

$$\frac{175 + 69 + 79}{3} \approx 107 \quad (2.6.17)$$

- *Maximum Intensity*. Con este tipo de proyección, cada píxel de la imagen de salida toma el nivel de brillo máximo de entre los niveles de brillo de los píxeles que tenían sus mismas coordenadas en las otras capas de la pila.

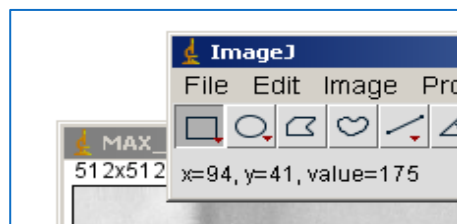


Figura 2.6.11. Nivel de brillo del píxel de coordenadas x=94 e y=41 en la proyección Z de tipo *Maximum Intensity* de la pila de prueba. Efectivamente, vale 175, que es el máximo nivel de brillo de los píxeles de las capas con esas coordenadas

- *Minimum Intensity*. Con este tipo de proyección, cada píxel de la imagen de salida toma el nivel de brillo mínimo de entre los niveles de brillo de los píxeles que tenían sus mismas coordenadas en las otras capas de la pila.

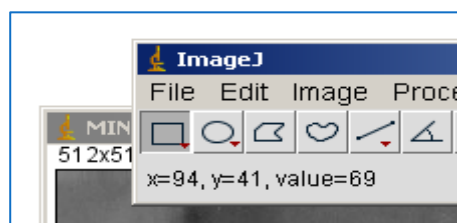


Figura 2.6.12. Nivel de brillo del píxel de coordenadas x=94 e y=41 en la proyección Z de tipo *Minumum Intensity* de la pila de prueba. Efectivamente, vale 69, que es el mínimo nivel de brillo de los píxeles de las capas con esas coordenadas

- *Sum Slices*. Con esta modalidad, la imagen de salida es el resultado de sumar directamente todas las capas de la pila (aunque para su representación se escala al margen dinámico de la pantalla).

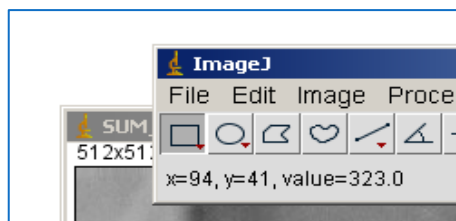


Figura 2.6.13. Nivel de brillo del píxel de coordenadas $x=94$ e $y=41$ en la proyección Z de tipo *Sum Slices* de la pila de prueba: 323

$$175 + 69 + 79 = 323 \quad (2.6.18)$$

- *Standard Deviation*. Con esta modalidad, el nivel de brillo de cada píxel de la imagen de salida es el la desviación estándar de los niveles de brillo de los píxeles que tenían sus mismas coordenadas en las otras capas de la pila:

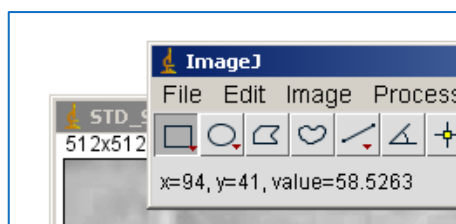


Figura 2.6.14. Nivel de brillo del píxel de coordenadas $x=94$ e $y=41$ en la proyección Z de tipo *Standard Deviation* de la pila de prueba: 58,5263

$$\sqrt{\frac{(107,667 - 175)^2 + (107,667 - 69)^2 + (107,667 - 79)^2}{3 - 1}} \approx 58,5263 \quad (2.6.19)$$

- *Median*. Con esta modalidad, el nivel de brillo de cada píxel de la imagen de salida es la mediana de los niveles de brillo de los píxeles que tenían sus mismas coordenadas en las otras capas de la pila:

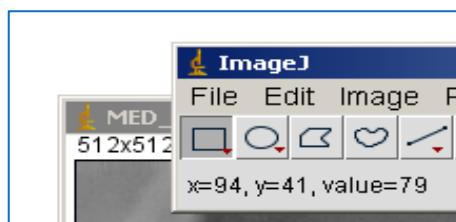


Figura 2.6.15. Nivel de brillo del píxel de coordenadas $x=94$ e $y=41$ en la proyección Z de tipo *Median* de la pila de prueba. Efectivamente, vale 79, que es la mediana de $\{69, 79, 175\}$

A continuación se muestran los resultados de hacer proyecciones z de diferentes tipos sobre la pila del archivo `t1-head.tif`, que contienen resonancias magnéticas de una cabeza humana:

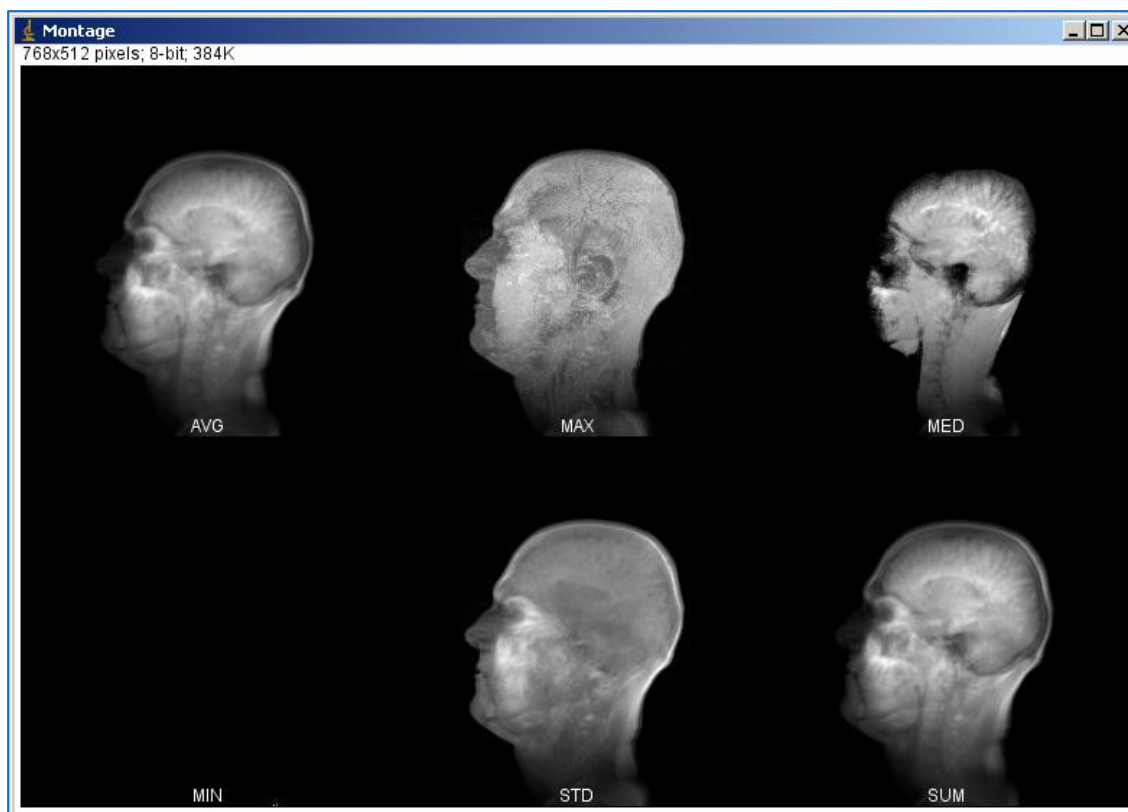


Figura 2.6.16. Proyecciones z de la pila del archivo t1-head.tiff. De izquierda a derecha y de arriba a abajo: *Average Intensity*, *Maximum Intensity*, *Median*, *Minimum Intensity*, *Standard Deviation* y *Sum Slices*

2.6.8. 3D Project

Si la herramienta anterior, *Z Project*, realizaba una proyección a lo largo del eje z, con la herramienta *3D Project* es posible realizar múltiples proyecciones desde diferentes ángulos de la información contenida en una pila, almacenar los resultados en una nueva pila y representar las capas de ésta en una secuencia animada de forma que se visualice un volumen rotatorio.

Para visualizar su funcionamiento hay que pensar en conjuntos de rayos (tantos como píxeles tenga cada proyección) paralelos entre sí y perpendiculares al eje de rotación que atraviesan el volumen de la pila y después chocan con los píxeles de un plano normal a su trayectoria, que será sobre el que se represente cada proyección, de forma que cada píxel pasará a tener un valor que dependerá de los valores de los píxeles que haya encontrado su correspondiente rayo en su recorrido a través del volumen de la pila. La forma en que se opera sobre estos valores de los píxeles encontrados en el recorrido para obtener el valor final depende del método de proyección elegido [6]. Estos, junto con otros muchos parámetros, se pueden determinar en el menú *3D Projection* (Figura 2.6.17):

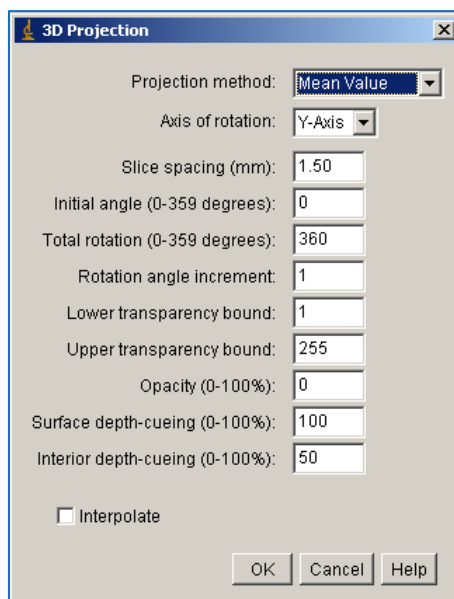


Figura 2.6.17. Menú *3D Projection*

- *Projection method*. Lista desplegable que permite elegir uno de los tres métodos de proyección disponibles:
 - *Nearest Point*. En cada proyección, cada píxel pasa a valer lo que vale el punto más cercano con valor no nulo que encuentra el rayo en su recorrido, de modo que este es el método ideal para ver las superficies externas del volumen contenido en la pila [6]. Se prueba con el archivo `t1-head.tiff` y varias de las proyecciones se muestran en la Figura 2.6.18:

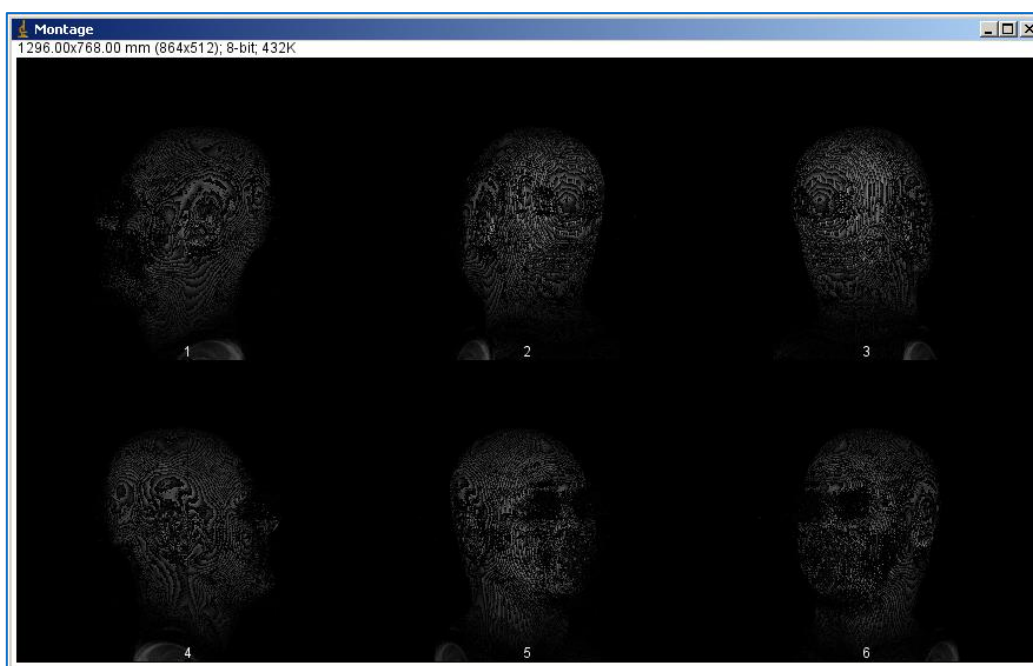


Figura 2.6.18. Varias proyecciones 3D de la pila del archivo `t1-head.tiff` con el método *Nearest Point*. Desde 0° a 359° cada 60° , tomando como no nulos aquellos vóxeles de brillo mayor que 1, con opacidad 0%, *Surface Depth-Cueing* de 100%, *Interior Depth-Cueing* de 50% y considerando todas las capas de la pila

- *Brightest Point*. Este método no considera los píxeles con valores no nulos más próximos, sino aquellos con mayores niveles de brillo que encuentran los rayos al atravesar el volumen de la pila en su camino hacia el plano de proyección. Este método es el adecuado para ver las partes más brillantes en el volumen, como los huesos en el caso de las tomografías computarizadas [6]. A continuación, en la Figura 2.6.19, se muestra un ejemplo del uso de este método:

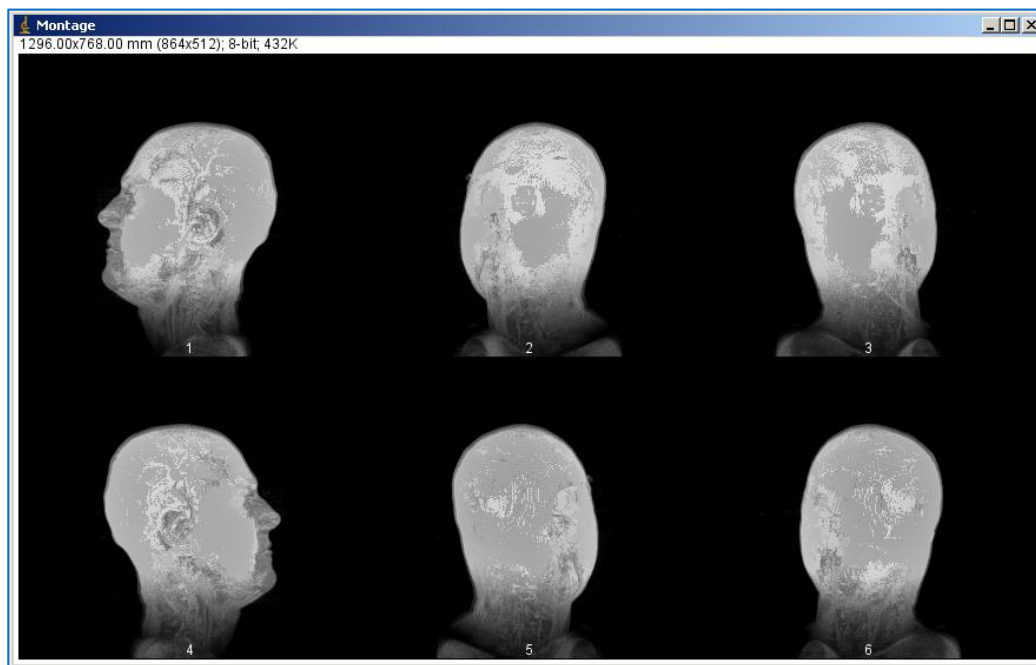


Figura 2.6.19. Varias proyecciones 3D de la pila del archivo t1-head.tiff con el método *Brightest Point*. Desde 0° a 359° cada 60°, tomando como no nulos aquellos vóxeles de brillo mayor que 1, con opacidad 0%, *Surface Depth-Cueing* de 100%, *Interior Depth-Cueing* de 50% y considerando todas las capas de la pila

- *Mean Value*. En este caso, los píxeles de cada proyección pasan a tomar por nivel de brillo la media de los niveles de brillo de los píxeles no nulos que sus correspondientes rayos han encontrado en su recorrido. Produce bordes más suavizados y un menor contraste, pero puede resultar útil a la hora de ver objetos contenidos en estructuras con un mayor nivel de brillo, como se aprecia en la Figura 2.6.20 [6].

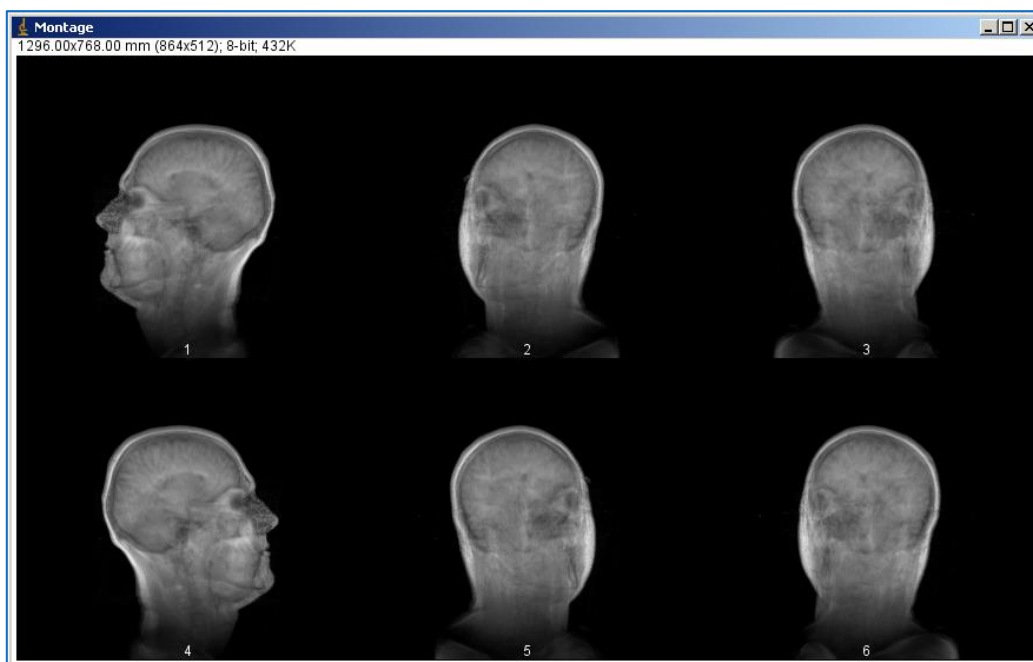


Figura 2.6.20. Varias proyecciones 3D de la pila del archivo `t1-head.tiff` con el método *Mean Value*. Desde 0° a 359° cada 60° , tomando como no nulos aquellos vóxeles de brillo mayor que 1, con opacidad 0%, *Surface Depth-Cueing* de 100%, *Interior Depth-Cueing* de 50% y considerando todas las capas de la pila

- *Axis of rotation.* En la Figura 2.6.18, en la Figura 2.6.19 y en la Figura 2.6.20 las proyecciones se han tomado desde diferentes ángulos rotando alrededor del eje y , pero también es posible tomar como eje de rotación el x o el z , y éste es el menú que permite hacerlo.
- *Slice spacing.* Permite definir cuántas capas de la pila se tienen en cuenta. Por defecto, este campo (espaciado entre capas) vale 1, con lo cual se tiene en cuenta todas las capas de la pila. Si este campo tomar un valor n , sólo se tendría en cuenta la primera capa de cada grupo de n capas. Evidentemente, cuantas más capas se tengan en cuenta, los resultados tienen mayor precisión, pero por otro lado el proceso supone una mayor carga computacional.
- *Initial angle.* Ángulo en grados en el que se empiezan a hacer las proyecciones.
- *Total rotation.* Ángulo en grados hasta el cual se hace las proyecciones.
- *Rotation Angle increment.* Incremento angular en grados que hay entre proyecciones consecutivas.
- *Lower/Upper transparency bound.* Límites superiores de transparencia. Permiten definir el intervalo de brillo (entre 0 y 255) que tienen los píxeles que se consideran no nulos de cara a los cálculos. Con una configuración adecuada, estos parámetros pueden ayudar a obtener una mejor visualización del volumen e incluso conseguir que se vean ciertas partes de interés y que desaparezcan las demás.
- *Opacity.* Cuando se elige el método *Brightest Point* o *Mean Value*, este parámetro permite ajustar la opacidad de forma que puedan llegar a distinguirse estructuras interiores a través superficies exteriores translúcidas. Lógicamente, cuanto mayor es la opacidad más se parecen los resultados a los que arrojaría

el método *Nearest Point* y por tanto las superficies externas son las que mejor se ven (de hecho, cuando el valor de *Opacity* es mayor que 0 se hace una combinación del método de proyección elegido con el método *Nearest Point*, en la que este último método tiene mayor peso en el resultado final cuanto mayor sea el parámetro *Opacity*). Por defecto vale 0, lo que significa que no se tiene en cuenta.

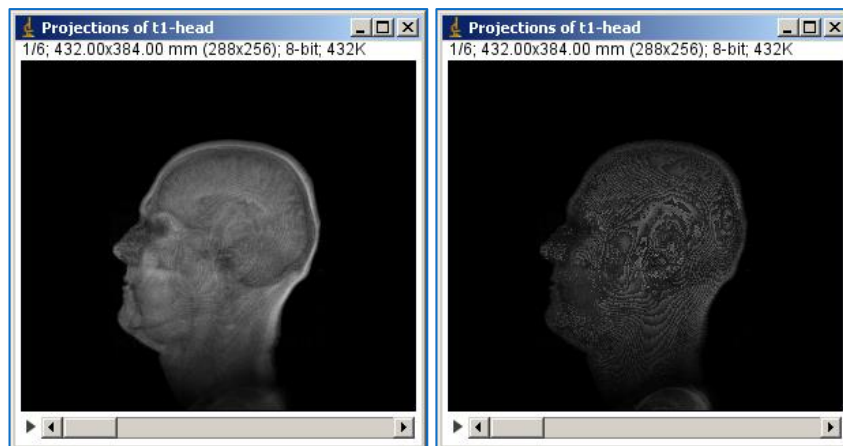


Figura 2.6.21. Izquierda: proyección con un valor del 25 % para el parámetro *Opacity* (se distinguen estructuras internas). Derecha: proyección con un valor del 75 % para el parámetro *Opacity* (apenas se distinguen estructuras internas, tan sólo las superficies externas)

- *Surface/Interior Depth-Cueing*. Con estos parámetros se puede modificar la profundidad de la imagen, de forma que los píxeles cuyo valor está determinado principalmente por vóxeles más cercanos al observador aparecen más brillantes, y este brillo se va atenuando de forma lineal conforme aumenta la distancia: si el parámetro toma un valor n mayor que 0, a los puntos más “cercaños” se les asigna el 100% de su nivel de brillo, a los más “lejanos” el $(100-n)\%$ de su nivel de brillo, y a los píxeles intermedios entre los dos extremos se les asigna un valor determinado linealmente y conforme a lo anterior. El parámetro *Surface Depth-Cueing* actúa sobre las proyecciones obtenidas con el método *Nearest Point* (y por tanto sobre la componente *Nearest Point* de la combinación de métodos que se lleva a cabo cuando se elige otro método y el parámetro *Opacity* toma un valor mayor que 0). El parámetro *Interior Depth-Cueing* actúa sobre las proyecciones obtenidas con el método *Brightest Point*. Ver Figura 2.6.22.
- *Interpolate*. En caso de que la pila tenga capas separadas más de un píxel (lo que empeoraría la calidad del resultado), al marcar esta opción se genera una nueva pila en la que las capas estén separadas tan solo por un píxel (a base de añadir capas interpolando las capas originales), y es esta nueva pila auxiliar la que sirve para hacer las proyecciones con más calidad.

Si previamente se realiza una selección sobre la pila, todo lo anterior se aplica a la parte de la pila seleccionada.

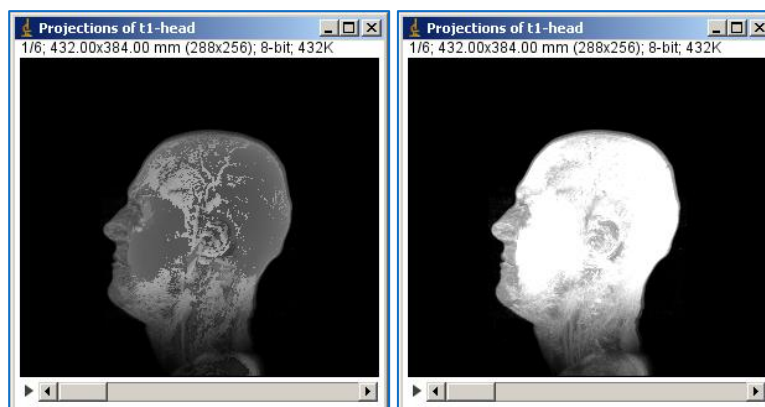


Figura 2.6.22. Izquierda: proyección de tipo *Brightest Point* con 100% de *Interior Depth-Cueing*. Derecha: misma proyección con 0% *Interior Depth-Cueing*. Se puede apreciar como en el primer caso, en el que se emplea un 100% de *Interior Depth-Cueing*, los píxeles que corresponden a zonas más lejanas tienen un nivel de brillo considerablemente menor, al contrario que en el segundo caso

2.6.9. Plot Z-axis Profile

Con esta herramienta es posible representar de forma gráfica la variación del nivel de brillo que tiene a lo largo del eje z una parte de la pila fija en el plano xy. Para determinar qué parte de la pila se considera basta con hacer una selección puntual o de área con las herramientas correspondientes. Si la selección es puntual, se representa de forma gráfica la variación de brillo a lo largo del eje z de todos los vóxeles que tengan las mismas coordenadas x e y que el punto seleccionado. Si la selección es de área, se representa la media de los niveles de brillo de los píxeles contenidos en dicha área en cada capa. A continuación, en la Figura 2.6.23, se muestra un ejemplo:

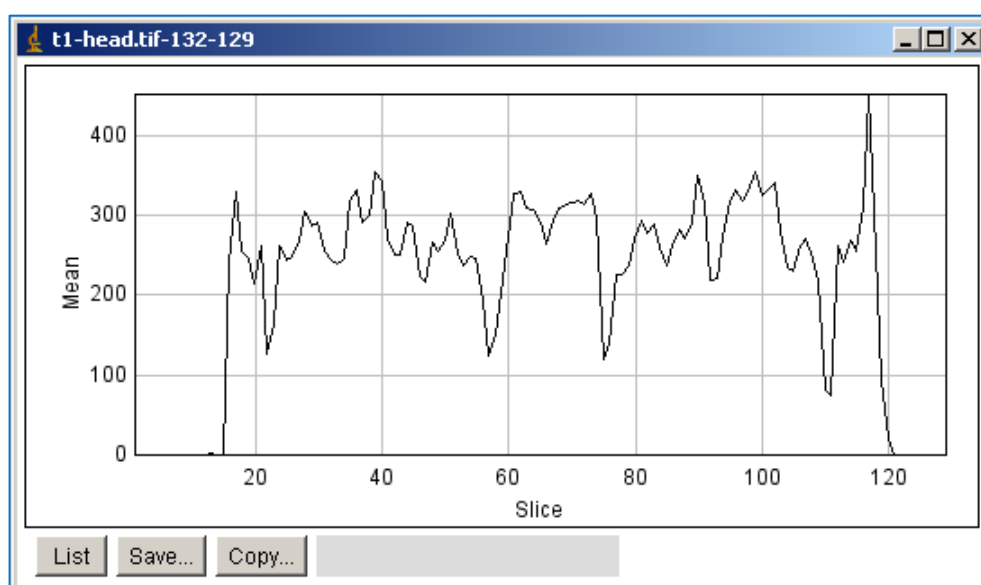


Figura 2.6.23. Niveles de brillo de los vóxeles con coordenadas x=198 mm e y=192 mm a lo largo del eje z en la pila correspondiente al archivo t1-head.tif

En la Figura 2.6.23 se muestra el resultado de aplicar la herramienta *Plot Z-axis Profile* tras haber realizado una selección puntual en la pila de t1-head.tiff en las coordenadas $x=198$ mm e $y=192$ mm. El eje de abscisas contiene 129 puntos, correspondientes a las 129 capas de la pila. En el eje de ordenadas se aprecian los niveles de brillo, que son mayores de 255 porque las capas de esta pila no son imágenes de tipo *8-bit* sino de tipo *16-bit*. La curva concuerda con lo esperado, teniendo en cuenta que las coordenadas $x=198$ mm e $y=192$ mm corresponden al centro de la cabeza: al principio y al final del eje z no hay vóxeles con brillo porque los correspondientes planos sagitales de la resonancia magnética corresponden a los hombros, pero entre ellos se empiezan a observar cambios en el nivel de brillo en función del tipo de materia del interior de la cabeza que atraviese cada capa.

Los datos en los que se basa el gráfico también se muestran formato numérico en una tabla que se puede editar y exportar al portapapeles del sistema.

2.6.10. Label

Seleccionando esta opción se despliega un menú que permite gestionar las etiquetas de las capas en una pila. Consta de los siguientes elementos:

- *Format*. Lista desplegable que permite especificar el formato de las etiquetas:
 - 0. Número aislado seguido de un espacio en blanco y de la cadena de caracteres fija especificada en el campo *Text*.
 - 0000. Número con ceros de relleno a la izquierda seguido de un espacio en blanco y de la cadena de caracteres fija especificada en el campo *Text*.
 - 00:00. Número en formato horario con dos niveles seguido de un espacio en blanco y de la cadena de caracteres fija especificada en el campo *Text*.
 - 00:00:00. Número en formato horario con tres niveles seguido de un espacio en blanco y de la cadena de caracteres fija especificada en el campo *Text*.
 - *Text*. Cadena de caracteres fija especificada en el campo *Text*.
 - *Label*. Nombre de la capa.
- *Starting Value*. Número desde el que se empieza a numerar las capas con los formatos numéricos.
- *Interval*. Intervalo entre los números de las etiquetas numéricas.
- *XY location*. Coordenadas x e y de la etiqueta.
- *Font size*. Tamaño de la etiqueta.
- *Range*. Rango de capas en las que se imprime la etiqueta.
- *Text*. Cadena de caracteres fija que aparece en los formatos numéricos y en el formato *Text*.

- *Use overlay*. Marcando esta casilla las etiquetas se añaden en forma de *Overlay* o recubrimiento (ver apartado 2.14 *Overlays*), de forma que las etiquetas se superponen a las capas sin modificar sus píxeles (de forma no invasiva) y se pueden activar y desactivar desde el submenú *Overlays* en el menú *Image*.
- *Use text tool font*. Marcando esta casilla se tienen en cuenta los parámetros especificados en el submenú *Fonts* (menú *Edit*) a la hora de imprimir las etiquetas.
- *Preview*. Vista previa.

2.6.11. Statistics

Seleccionando esta opción se analiza la pila y se elabora un resumen de sus estadísticas que se presenta en la tabla de resultados. Se muestra la siguiente información: número de vóxeles, media de nivel de brillo, desviación estándar del nivel de brillo, nivel de brillo mínimo y nivel de brillo máximo.

2.6.12. Tools

En este último submenú se agrupan las últimas herramientas para el manejo de pilas:

- *Combine*. Esta herramienta permite combinar dos pilas en una nueva pila. Al seleccionar la opción se abre un menú que permite elegir las pilas que se van a combinar y el modo en que se combinan. Con la opción *Combine vertically* desmarcada, la nueva pila tiene una anchura equivalente a la suma de las anchuras de las pilas originales (puesto que las capas de una pila se presentan en horizontal al lado de las de la otra), la altura máxima de las alturas de las pilas originales y la profundidad máxima de las profundidades de las pilas originales. Con la opción *Combine vertically* marcada, la nueva pila tiene por anchura la anchura máxima de las pilas originales, una altura equivalente a la suma de las alturas de las pilas originales (puesto que las capas de una pila se presentan en vertical junto a las de la otra) y la profundidad máxima de las profundidades de las pilas originales.
- *Concatenate*. Concatena imágenes y pilas que coincidan en tipo y dimensiones. Además, si se concatenan pilas con la misma profundidad y se marca la opción *Open as 4D image* el resultado es una hiperpila de 4 dimensiones de forma que las pilas originales se distribuyen a lo largo del eje temporal.
- *Reduce*. Permite diezmado las pilas en su dimensión z por el factor de diezmado especificado en el campo *Reduction Factor* del menú emergente (un factor de diezmado n implica eliminar una capa de cada grupo de n capas). También funciona con hiperpilas, y en su caso el diezmado se realiza en la dimensión temporal a no ser que esté marcada la casilla *Reduce in Z-Dimension*, en cuyo caso también se diezma la hiperpila en la dimensión z.
- *Reverse*. Lleva a cabo un *flip* en el eje z (la primera capa pasa a ser la última, la segunda pasa a ser la penúltima y así sucesivamente).

- *Insert*. Inserta una pila (*Source*) en otra (*Destination*) a partir de las coordenadas x e y especificadas (cada capa de la pila fuente se superpone a la correspondiente capa de la pila destino a partir de las coordenadas indicadas). Funciona también con imágenes individuales.
- *Montage to Stack*. Al contrario de lo que hace la herramienta *Make Montage*, *Montage to Stack* construye una pila a partir de un montaje teniendo en cuenta las columnas, las filas y el ancho de borde indicado en el menú emergente.
- *Make Substack*. Extrae capas de una pila y con ellas construye una subpila en una nueva ventana. En el menú emergente se puede determinar qué capas se consideran especificando intervalos de capas, intervalos de capas con incrementos o listas de capas.
- *Grouped Z Project*. Hace proyecciones a lo largo del eje z de grupos de capas de una misma pila. Al seleccionar la opción aparece un menú que permite seleccionar el método de proyección (ver el apartado 2.6.7. *Z Project*), así como el tamaño de los grupos (*Group size*). Este parámetro no puede tomar cualquier valor, sino números divisores del número total de capas de la pila original. Las proyecciones resultantes se agrupan en una nueva pila.
- *Set Label*. Permite determinar de forma manual la etiqueta de cada capa de la pila.
- *Remove Slice Labels*. Elimina las etiquetas de las capas de las pilas.
- *Start/Stop Animation*. Activa y desactiva la reproducción de las capas de la pila una tras otra con el *frame rate* especificado en el menú *Animation Options*.
- *Animation Options*. Despliega un menú que permite modificar los parámetros relacionados con la animación de las capas de las pilas:
 - *Speed*. Velocidad de la reproducción en fps.
 - *First/Last Frame*. Primera y última capa que se reproduce.
 - *Loop Back and Forth*. Marcando esta casilla se activa la reproducción en bucle.
 - *Start Animation*. Marcando esta casilla se activa la reproducción al salir del menú.

2.7. Submenú Hyperstacks

En este submenú se agrupan las herramientas que se aplican de forma exclusiva a las hiperpilas. Si las pilas, estudiadas en el menú anterior, se entendían como conjuntos de imágenes relacionadas en la tercera dimensión, de forma que constituirían un volumen, las hiperpilas son pilas que se relacionan a lo largo del tiempo (en una cuarta dimensión) y que se pueden administrar a través de canales (en una quinta dimensión), por lo que se pueden entender como conjuntos de imágenes de cinco dimensiones.

2.7.1. New Hyperstack.

Esta opción despliega un menú que permite crear una nueva pila especificando sus características fundamentales:

- *Title*. Título de la hiperpila.
- *Type*. Lista desplegable que permite elegir el tipo de las imágenes que formarán la hiperpila. Las opciones son *8-bit*, *16-bit*, *32-bit* y *RGB Color* (ver apartado 2.1. *Submenú Type*).
- *Display Mode*. Lista desplegable que permite elegir la forma en que se representa la hiperpila. Las opciones son las siguientes:
 - *Composite*. Las imágenes se muestran en modo compuesto y es posible ver al mismo tiempo todos los canales que se deseen. Para activar o desactivar canales sólo hay que marcar o desmarcar sus correspondientes casillas en el menú *Channels Tool*.
 - *Color*. Sólo se muestra un canal a la vez, y se muestra coloreado según el color al que corresponda. Para elegir el canal que se muestra sólo hay que marcar su casilla en el menú *Channels Tool*.
 - *Grayscale*. Sólo se muestra un canal a la vez, y se muestra en escala de grises (es decir, la intensidad del color se representa mediante el nivel de brillo). Para elegir el canal que se muestra sólo hay que marcar su casilla en el menú *Channels Tool*.
- *Width*. Permite especificar de forma numérica el ancho de las imágenes de la hiperpila.
- *Height*. Permite especificar de forma numérica la altura de las imágenes de la hiperpila.
- *Channels (c)*. Permite especificar el número de canales de la hiperpila.
- *Slices (z)*. Permite especificar el número de capas de la hiperpila.
- *Frames (t)*. Permite especificar el número de cuadros de la hiperpila.
- *Label Images*. Marcando esta casilla, a cada imagen de la hiperpila se la marca con su correspondiente número de capa, canal y cuadro.

Al crear la hiperpila, sus imágenes aparecen vacías (en negro).

2.7.2. Stack to Hyperstack.

Esta opción permite convertir pilas en hiperpilas. A través de un menú emergente es posible especificar el papel que pasan a tomar en la hiperpila las capas de la pila: orden relativo entre canales, capas y cuadros, número de canales, número de capas, número de cuadros y método de representación. Sin embargo, hay que tener en cuenta que sea cual sea la forma en que se distribuyen, el número de imágenes en la pila y en la hiperpila debe ser el mismo, y en caso de que no sea así se muestra un mensaje de error. También cabe mencionar que si la pila es de tipo *RGB Color*, la hiperpila resultante tendrá tres canales: R, G y B.

2.7.3. Hyperstack to Stack.

Operación inversa a la anterior: convierte una hiperpila en una pila, y por defecto lo hace en orden *czt* (canales, capas y cuadros), lo cual significa que la primera capa de la nueva pila será la imagen que antes era la primera capa del primer cuadro en el primer canal, la segunda capa de la nueva pila será la imagen que antes era la primera capa del primer cuadro en el segundo canal, y así sucesivamente.

2.7.4. Reduce Dimensions.

Permite reducir las dimensiones de la hiperpila activa. Para ello se despliega un menú con casillas que corresponden a los canales (*Channels*), las capas (*Slices*) y los cuadros (*Frames*). Desmarcando, por ejemplo, la casilla correspondiente a los canales, se eliminan todos los canales salvo el activo, y lo mismo ocurre para las otras dos dimensiones. La casilla *Keep source* permite conservar la pila original después de la operación.

2.8. Crop

Esta herramienta permite recortar una selección rectangular hecha sobre una imagen y mostrarla en una nueva ventana. No sólo funciona con imágenes sencillas, sino también con pilas e hiperpilas, que pasan a tener la anchura y la altura de la selección rectangular, pero que siguen siendo pilas e hiperpilas y conservan el resto de sus dimensiones. Además funciona con otro tipo de selecciones: por ejemplo, si se hace una selección elíptica, la nueva imagen tendrá las dimensiones del rectángulo más pequeño que se pueda circunscribir a dicha elipse.



Figura 2.8.1. Izquierda: imagen original con selección elíptica. Derecha: imagen recortada según la selección anterior

2.9. Duplicate

Permite duplicar imágenes, pilas e hiperpilas y abrirlas en una nueva ventana. En el caso de las pilas e hiperpilas, permite especificar a través de un menú el rango de canales, capas y cuadros que se duplican.

También funciona con selecciones, de modo que en la nueva ventana aparece la selección pero se conserva la imagen original en otra ventana.

2.10. Rename

Permite establecer el nombre de la imagen activa.

2.11. Scale

Permite redimensionar la imagen, pila o hiperpila activa. Despliega un menú que contiene los siguientes elementos:

- *XY/Z Scale*. Son los factores por los que se escalan las dimensiones de la imagen. Por ejemplo, el ancho de la nueva imagen será $X \text{ Scale} \cdot \text{ancho original}$. Al fijar estos parámetros quedan determinados de forma automática los parámetros *Width*, *Height* y *Depth*.
- *Width/Height/Depth*. Permiten especificar directamente las dimensiones de la nueva imagen.
- *Interpolation*. El redimensionamiento de la imagen conlleva realizar interpolaciones para calcular los niveles de brillo de aquellos píxeles que no existían en la imagen original. Esta lista desplegable permite elegir el tipo de interpolación: bilineal (*Bilinear*) o bicúbica (*Bicubic*).
- *Fill with background color*. Si se redimensiona una selección de área y no se va a crear una nueva ventana, marcando esta casilla se rellenarán los huecos que aparezcan con el color de fondo especificado en el menú *Color Picker*. En caso contrario, se rellenan con negro.
- *Average when downsizing*. Como ya se ha comentado, al marcar esta casilla y reducir las dimensiones de las imágenes se lleva a cabo de forma previa un filtrado paso bajo de media para prevenir la aparición de *aliasing* que se puede derivar de la reducción de tamaño (que al fin y al cabo es un submuestreo).
- *Process entire stack*. Marcando esta casilla, las pilas e hiperpilas se escalan también en la dimensión z según lo indicado en el campo correspondiente.
- *Create new window*. Marcando esta casilla la imagen redimensionada se lleva a una nueva ventana (cuyo nombre se puede especificar en el campo *Title*).

2.12. Submenú Transform

En este submenú se agrupan diversas herramientas que permite realizar principalmente transformaciones geométricas sobre las imágenes.

2.12.1. Flip Horizontally

Lleva a cabo un *flip* horizontal, de modo que, en cada fila, el último píxel pasa a ser el primero, el penúltimo pasa a ser el segundo, y así sucesivamente. La Figura 2.12.1 ilustra este concepto:

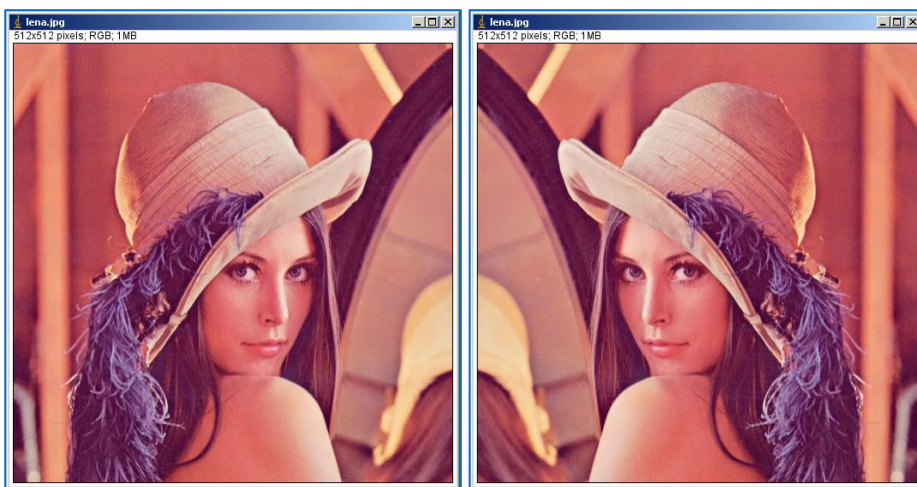


Figura 2.12.1. Izquierda: lena.jpg. Derecha: lena.jpg tras un *flip* horizontal

2.12.2. Flip Vertically

Lleva a cabo un *flip* vertical, de modo que, en cada columna, el último píxel pasa a ser el primero, el penúltimo pasa a ser el segundo, y así sucesivamente. La Figura 2.12.2 ilustra este concepto:

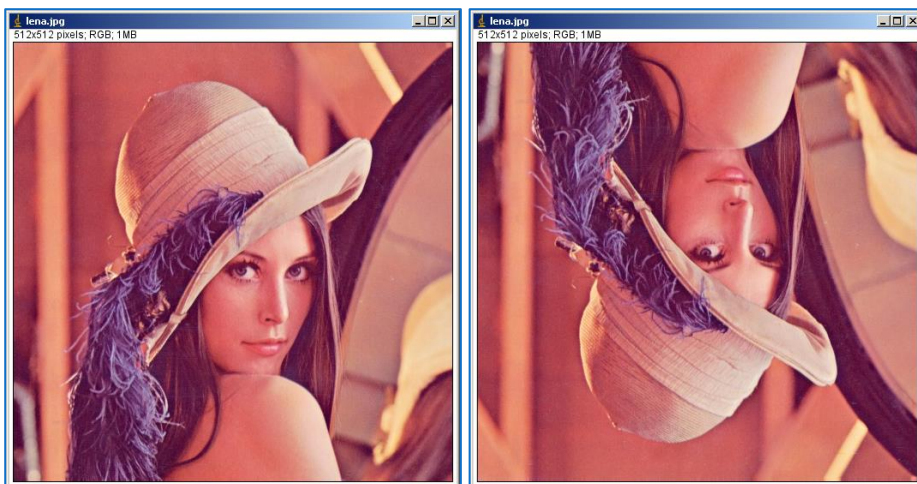


Figura 2.12.2. Izquierda: lena.jpg. Derecha: lena.jpg tras un *flip* vertical

2.12.3. Flip Z

En caso de que se aplique sobre una pila o hiperpila, lleva a cabo un *flip* en el eje z, de modo que la última capa pasa a ser la primera, la segunda pasa a ser la segunda, y así sucesivamente.

2.12.4. Rotate 90 Degrees Right/Left

Gira la imagen, pila o hiperpila 90° a la derecha o a la izquierda.

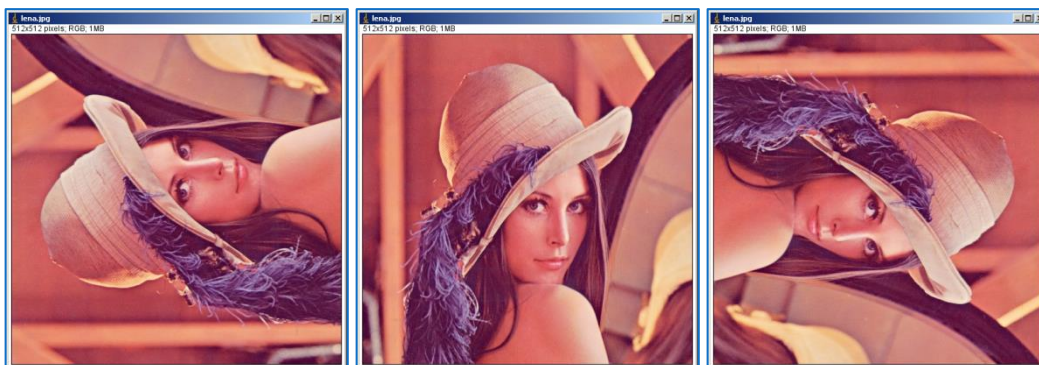


Figura 2.12.3. Izquierda: lena.jpg girada 90° a la izquierda. Centro: lena.jpg. Derecha: lena.jpg girada 90° a la derecha

2.12.5. Rotate

Permite girar la imagen (o selección) en sentido horario un número de grados determinado por el usuario (para que el giro se realice en sentido anti horario basta con especificar un ángulo negativo). En la Figura 2.12.4 se muestra un ejemplo:

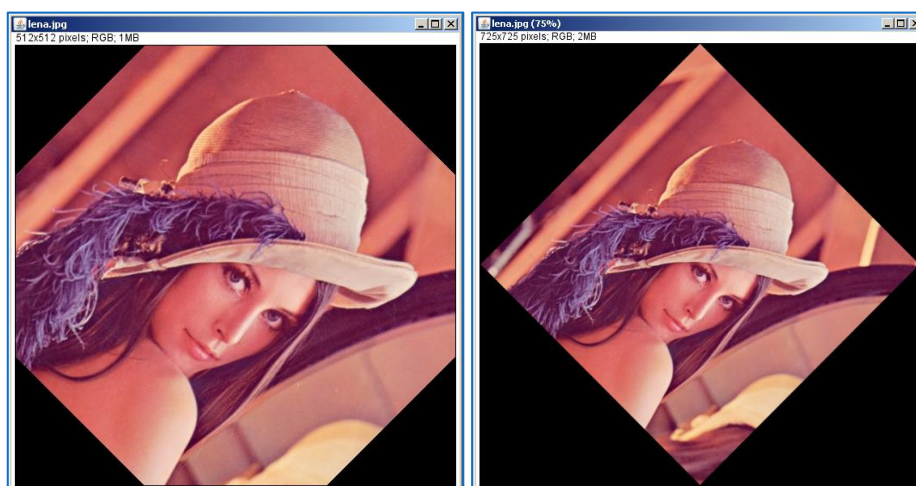


Figura 2.12.4. Izquierda: lena.jpg girada 45° sin marcar *Enlarge Image to Fit Results*. Derecha: lena.jpg girada 45° marcando *Enlarge Image to Fit Results*

Despliega un menú que permite especificar varios parámetros:

- *Angle*. Grados del ángulo que ha de girar la imagen.
- *Grid Lines*. En el modo vista previa, permite especificar el número de guías que se imprimen: un valor n implica n guías horizontales y verticales equiespaciadas.
- *Interpolation*. Los giros implican la aparición de nuevos píxeles en nuevas posiciones, cuyo valor hay que calcular a base de interpolar con respecto a los píxeles adyacentes. Esta lista desplegable permite especificar el tipo de interpolación que se emplea: bilineal (*Bilinear*) o bicúbica (*Bicubic*), que ya se han comentado en el apartado 6.2. *Size*.
- *Fill with Background Color*. Evidentemente, al girar una imagen rectangular es posible que aparezcan huecos en la ventana que la contiene. Por defecto, a los píxeles que componen dichos huecos se le asigna

un nivel de brillo 0 (se rellenan de negro), pero marcando esta casilla los huecos se rellenan con el color de fondo elegido en el menú *Color Picker*.

- *Enlarge Image to Fit Results*. Marcando esta casilla, la ventana que contiene la imagen girada es lo suficientemente grande para albergar toda la imagen. En caso contrario, la nueva ventana tiene el mismo tamaño que la original, y por tanto es posible que se pierdan las esquinas de la imagen al girarla.
- *Preview*. Habilita la vista previa.

2.12.6. Translate

Permite mover la imagen horizontalmente el número de píxeles especificado en el campo *X Offset*, y verticalmente el número de píxeles especificado en el campo *Y Offset*. *Offsets* positivos conllevan desplazamientos hacia la derecha y hacia abajo, y *offsets* negativos conllevan desplazamientos hacia la izquierda y hacia arriba. Una vez más, se puede especificar el método de interpolación: bilineal o bicúbica. Los huecos que aparecen al desplazar la imagen se rellenan con píxeles de brillo 0.

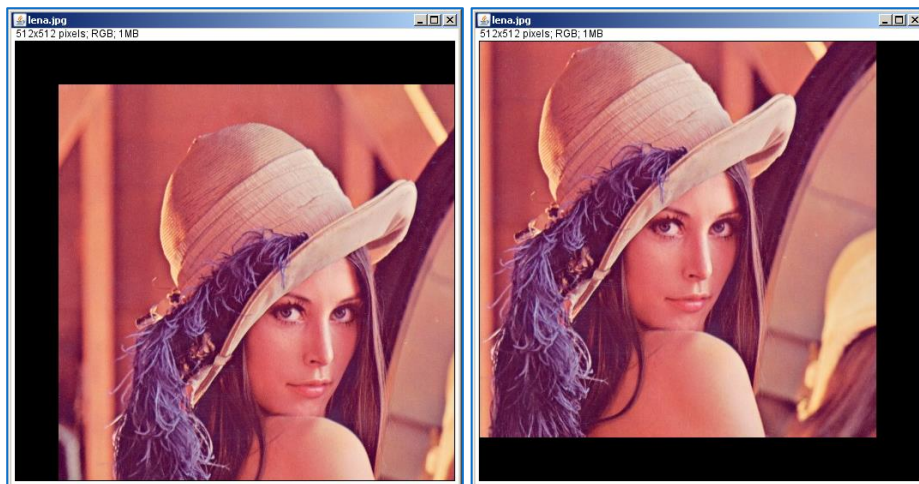


Figura 2.12.5. Izquierda: lena.jpg con *X Offset*=50 e *Y Offset*=50. Derecha: lena.jpg con *X Offset*=-50 e *Y Offset*=-50

2.12.7. Bin

Reduce el tamaño de las imágenes a base de compactar grupos de píxeles. Un menú emergente permite elegir el factor de reducción en la dimensión *x* (*X shrink factor*) y en la dimensión *y* (*Y shrink factor*), y además permite elegir el método de compactación. Por cada grupo de píxeles de dimensiones *X shrink factor* x *Y shrink factor*, la nueva imagen reducida tendrá un píxel cuyo valor será la media, la mediana, el máximo o el mínimo de los valores de los píxeles originales de dicho grupo, dependiendo de si el método de compactación es *Average*, *Median*, *Maximum* o *Minimum* respectivamente. También es posible compactar pilas e hiperpilas especificando el factor de reducción en la dimensión *z* (*Z shrink factor*). Esta última operación produce resultados similares a los de la herramienta *Grouped Z Project*.

2.12.8. Image to Results

Con esta herramienta, la selección de área, o en su defecto la imagen activa completa, se traslada a la tabla de resultados de forma que en la celda de coordenadas x_0, y_0 (las coordenadas se detallan en los bordes de la tabla) aparece el valor numérico del nivel de brillo del píxel de coordenadas x_0, y_0 . Esto se muestra en la Figura 2.12.6:

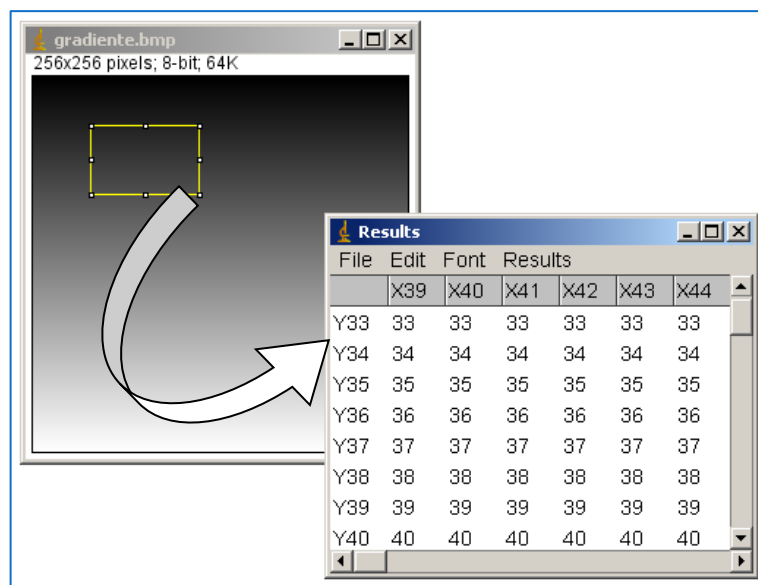


Figura 2.12.6. Imagen gradiente.bmp con una selección rectangular junto al contenido de dicha selección traducido a valores numéricos en la tabla de resultados.

En la figura se observa una imagen de un degradado de gris en la que el nivel de brillo aumenta con la coordenada y . Por ello, en la tabla de resultados se aprecia que los valores numéricos aumentan fila a fila. El nivel de brillo de los píxeles de aquellas imágenes que estén calibradas o que usen representación de punto flotante tiene tanta precisión como se especifique en el campo *Decimal Places* en el submenú *Measurements*, menú *Analyze*. Por otro lado, en imágenes en color el valor numérico de su nivel de brillo se calcula haciendo la media aritmética sin ponderar de los niveles de rojo, verde y azul, a no ser que en el menú *Conversions* (submenú *Options* del menú *Edit*) se marque la opción *Weighted RGB to Grayscale Conversion*, en cuyo caso se utiliza la expresión 2.1.3.

2.12.9. Results to Image

Esta herramienta lleva a cabo la operación inversa a la anterior: el contenido numérico de la tabla de resultados se convierte en imagen (de tipo *32-bit*) haciendo corresponder al píxel de coordenadas x_0, y_0 el nivel de brillo especificado de forma numérica en la celda de coordenadas x_0, y_0 de la tabla de resultados.

2.13. Submenú Zoom

En este submenú se agrupan los comandos relacionados con el *zoom*.

- *In.* Lleva a cabo un *zoom in* sobre la imagen (hay disponibles 21 aumentos predeterminados que alcanzan hasta 3200%)
- *Out.* Lleva a cabo un *zoom out*.
- *Original Scale.* Muestra la imagen con el aumento original.
- *View 100%.* Muestra la imagen de forma que un píxel de la imagen se hace corresponder a un píxel de la pantalla.
- *To Selection.* Hace un *zoom* en función de la selección: se elige de entre los aumentos predeterminados el más grande con el que se pueda mostrar la selección al completo sin que se salga de la ventana.
- *Set.* Permite determinar manualmente el aumento, que no tiene por qué ser uno de los predeterminados. Además, permite especificar el punto en el que se centra el *zoom*.

2.14. Submenú Overlay

Este menú contiene las herramientas necesarias para administrar los recubrimientos (*overlays*). El recubrimiento de una imagen es una capa imaginaria superpuesta a la imagen en la que caben todo tipo de selecciones (hasta texto), y que se caracteriza por ser no destructiva: su contenido no modifica el valor de los píxeles de la imagen. Con las siguientes opciones es posible añadir y quitar elementos del recubrimiento, mostrar u ocultar el recubrimiento, etiquetarlo, etc. Si una imagen con recubrimiento se guarda en formato TIFF, su recubrimiento se conserva.

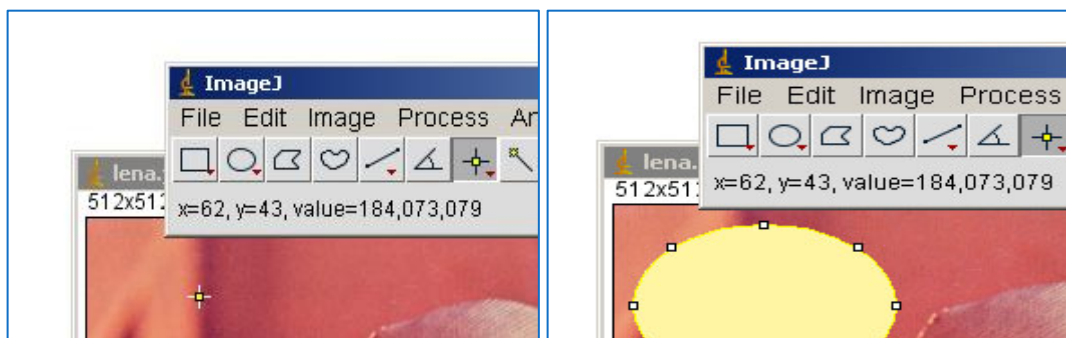


Figura 2.14.1. Izquierda: fragmento de una imagen; en la barra de estado se destaca que el valor del píxel de coordenadas $x=62$ e $y=43$ tiene valores $R=184$, $G=073$ y $B=079$. Derecha: mismo fragmento de la imagen anterior con un recubrimiento que lo cubre en gran parte; a pesar de ello, se puede apreciar que el píxel de coordenadas $x=62$ e $y=43$, que está debajo del recubrimiento, sigue teniendo los mismos valores RGB.

- *Add Selection.* Añade al recubrimiento la selección actual. También se puede hacer pulsando la tecla “B” directamente. La combinación “Alt” + “B” abre el menú *Add to Overlay*, explicado más adelante.
- *Add Image.* Permite añadir una imagen al recubrimiento de la imagen activa de forma que, en la práctica, las dos imágenes se mezclan. A través de un menú emergente es posible determinar la opacidad de la

imagen que se añade (en el campo *Opacity*), así como su localización (la de la esquina superior izquierda) sobre la imagen de fondo.

- *Hide Overlay*. Esta opción oculta el recubrimiento de la imagen activa en caso de que se estuviera mostrando.
- *Show Overlay*. Esta opción muestra el recubrimiento de la imagen activa en caso de que se estuviera ocultando.
- *From ROI Manager*. Elimina el recubrimiento previo y crea un nuevo recubrimiento con las selecciones almacenadas en el *ROI Manager* o Gestor de Regiones de Interés (ver el apartado correspondiente a *ROI Manager*).
- *To ROI Manager*. Traslada al Gestor de Regiones de Interés los elementos que componen el recubrimiento, eliminando los elementos que había previamente en el Gestor de Regiones de Interés.
- *Remove Overlay*. Elimina de forma permanente el recubrimiento actual (no se restaurará aunque se seleccione la opción *Show Overlay*).
- *Flatten*. Esta opción crea una nueva ventana con una imagen de tipo *RGB Color* del mismo tamaño que la imagen activa que es el resultado de imprimir de forma definitiva el recubrimiento sobre dicha imagen. Aunque es cierto que se conserva la imagen original en otra ventana, la operación deja de ser no destructiva, puesto que los píxeles de la imagen se modifican para incluir el recubrimiento, que deja de ser una capa imaginaria asociada a ella.
- *Labels*. Muestra un menú emergente que permite gestionar las etiquetas de los recubrimientos. Contiene los siguientes elementos:
 - *Color*. Lista desplegable que permite elegir el color de las etiquetas entre varios colores predeterminados.
 - *Font size*. Tamaño de las etiquetas.
 - *Show labels*. Si se marca esta casilla, se muestra una etiqueta junto a cada elemento del recubrimiento (por defecto, un número).
 - *Draw backgrounds*. Si se marca esta casilla, la etiqueta se muestra sobre un fondo coloreado con el color complementario al de la etiqueta.
 - *Bold*. Muestra la etiqueta en negrita.
- *Overlay Options*. Permite especificar el color del trazo en forma de cadena de caracteres o valor hexadecimal (*Stroke color*) y su ancho (*Width*), y en caso de que se trate de texto, también se puede especificar su tamaño y su justificación. Marcando la casilla *New overlay* se deshecha el recubrimiento anterior y la selección actual se añade a un nuevo recubrimiento

2.15. Submenú Lookup Tables

En este submenú se administran las LUTs (*Lookup Tables*), de las cuales ya se ha hablado anteriormente. Cuando se selecciona, se despliega una lista con todas las tablas LUT que hay en el sistema, que son las que incorpora por defecto ImageJ más todas aquellas instaladas de forma manual (basta con situar sus correspondientes ficheros en el directorio ... */ImageJ/luts*). Eligiendo una de estas tablas se aplica a la imagen activa, siempre que sea una imagen en escala de grises o de tipo *8-bit Color*. Con esto se consigue una imagen pseudocoloreada: a cada nivel de gris en la imagen se le asocia el color de la tabla cuyo índice coincide numéricamente con dicho nivel de gris. Eligiendo la tabla adecuada, esta técnica permite poner de manifiesto pequeñas variaciones de gris que normalmente no se detectarían sin color, y que sin embargo contienen información relevante para estudios médicos, astronómicos, geológicos,...Esta ventaja se aprecia en la Figura 2.15.1:

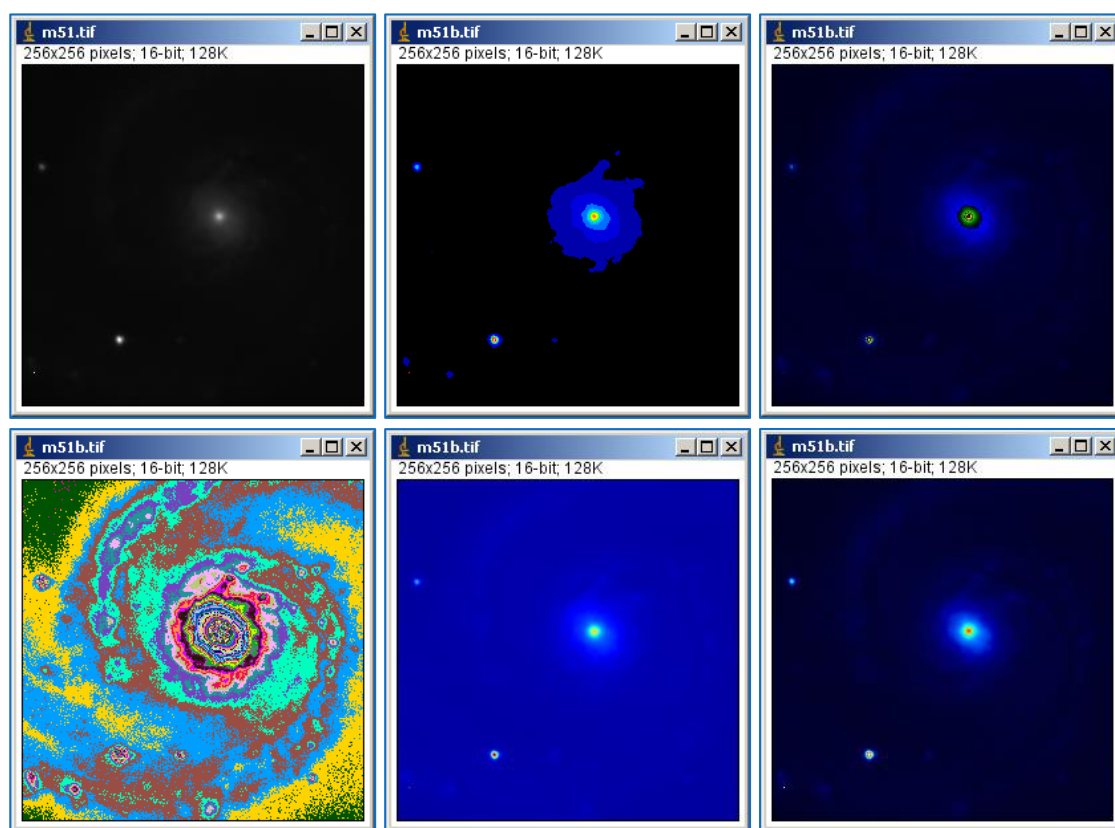


Figura 2.15.1. De izquierda a derecha y de arriba abajo: imagen original, m51.tiff (vista de la galaxia Objeto Messier 51); imagen original con la LUT 16 Colors; imagen original con la LUT 5 Ramps; imagen original con la LUT Glasbey; imagen original con la LUT Jet; imagen original con la LUT Royal.

En la parte superior izquierda de la Figura 2.15.1 se aprecia una imagen de la galaxia Objeto Messier 51 de 16 bits de profundidad, lo que implica que tiene una gran cantidad de niveles de gris que, sin embargo, no se podrían percibir aunque se representaran todos (lo cual no es el caso, pues en la pantalla se representan 256 niveles). Sin embargo, resulta más fácil distinguir las diferencias en la imagen cuando se colorea, como se

puede apreciar en las otras cinco imágenes de la figura, que son el resultado de aplicar diferentes LUTs a la imagen original.

3. MENÚ PROCESS

En este menú se agrupan las herramientas directamente relacionadas con el procesado de imagen: principalmente, operaciones de punto, operaciones aritméticas y filtrado, tanto en el dominio espacial como en el dominio de la frecuencia espacial. En la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación de la Universidad Politécnica de Madrid ya se han llevado a cabo varios proyectos directamente relacionados con los conceptos abarcados en este tema, y también se han desarrollado varios programas propio de la institución que ofrecen herramientas muy similares a las que se agrupan en el menú *Process* de ImageJ. Por todo lo anterior, la revisión de esta parte del programa no es tan exhaustiva como en el resto de los casos. El principal propósito de la inclusión de este tema es completar el análisis global de ImageJ.

3.1. Smooth

Con esta herramienta se suaviza la imagen activa con un filtrado espacial suavizante prefijado: emplea un filtro de media con una máscara 3x3. Este filtro, por ser espacial, provoca que en la imagen procesada cada píxel pase a tomar un valor que depende del nivel de brillo que tenía el propio píxel en la imagen original y también de los valores que tenían los píxeles de su entorno. Por ser suavizante, este filtro tiene carácter paso bajo, por lo que respeta las frecuencias espaciales bajas y atenúa las altas. Estas frecuencias espaciales altas corresponden a los detalles finos (en los que los niveles de brillo varían de forma muy rápida en incrementos espaciales pequeños), y por lo tanto el efecto de este filtro es emborronar o suavizar la imagen, y de ahí su nombre. Por último, por ser este filtro espacial suavizante de media, se implementa a base de reemplazar el nivel de brillo de cada píxel por la media de su valor y los valores de los píxeles de su entorno [8]. Una forma de ilustrar esto es pensar en una plantilla o máscara como la de la Tabla 3.1.1 que se va desplazando por cada uno de los píxeles haciendo coincidir su celda central con ellos y las exteriores con sus píxeles adyacentes, y dando a cada píxel el peso en la media que contenga la celda superpuesta.

Tabla 3.1.1. Máscara de un filtro de media especial 3x3

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

La operación descrita anteriormente se puede implementar, con ciertas precauciones, como una convolución bidimensional de la imagen con la máscara.

Un ejemplo del efecto de este filtro se muestra en la Figura 3.1.1. En ella se observa un detalle de la imagen original lena.jpg (en la izquierda), y el mismo detalle tras haberle aplicado la herramienta *Smooth* una vez (en

el centro) y tras haberle aplicado la herramienta *Smooth* dos veces (derecha). Es evidente el progresivo emborronamiento o suavizado de la imagen.

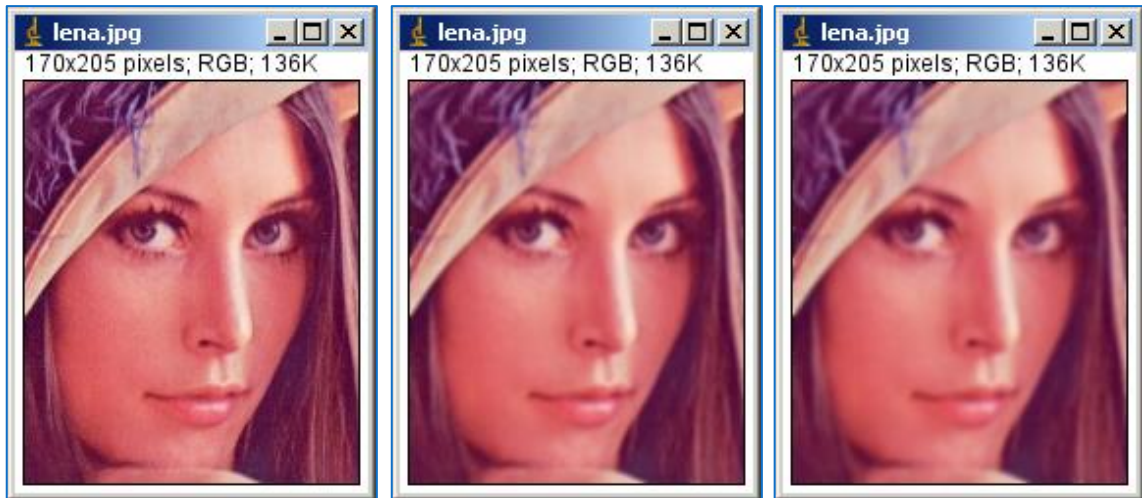


Figura 3.1.1. Detalle de lena.jpg. De izquierda a derecha: original, tras aplicar *Smooth* una vez y tras aplicar *Smooth* dos veces

Para comprobar que esta herramienta se ha implementado como se explicaba anteriormente, se comparan los valores del brillo⁴ (que es la característica sobre la que se opera) de la imagen antes y después de suavizarla:

Results					Results				
File	Edit	Font	Results		File	Edit	Font	Results	
X58	X59	X60	X61		X58	X59	X60	X61	
179.667	169	137.667	139.667		175.667	159.667	147	149	
181.333	173	149.667	147.333		179	167.333	157	157.667	
177.333	177.333	162.333	160		181	172.667	166	166.667	
188	178.333	173.333	178.333		181.333	176.333	171.667	173.333	
185.333	177.333	172.333	171.333		181.667	178.667	176.667	177.333	
178.667	179.667	180	181.333		181.667	178.667	177.333	178.667	
184	178.333	179	181.333		182.667	180.667	180	177	
185	184.333	182.667	178.667		185.333	182	175.333	161.333	

Figura 3.1.2. Valores del brillo de la imagen lena.jpg antes (izquierda) y después (derecha) de suavizarla con la herramienta *Smooth*

En la parte izquierda de la Figura 3.1.2 se destacan en rojo los niveles de brillo de un píxel y su entorno, los cuales forman juntos una matriz de píxeles de tamaño 3x3. Haciendo la media de todos ellos resulta lo siguiente:

$$\frac{1}{9}(178,333 + 173,333 + \dots + 180 + 181,333) \approx 176,666 \quad (3.1.1)$$

⁴ Obtenido con la ecuación (2.1.3)

Como se puede observar, la media es el nivel de brillo que toma el mismo píxel en la imagen procesada. Precisamente por este comportamiento, este tipo de filtros contribuyen a eliminar el ruido aleatorio de imágenes ruidosas, asociando a los píxeles afectados por el ruido valores acordes a su entorno, que en la mayoría de los casos (y si el ruido no es excesivo) no estará completamente afectado por el ruido. Un ejemplo de esta aplicación de esta herramienta se muestra a continuación:

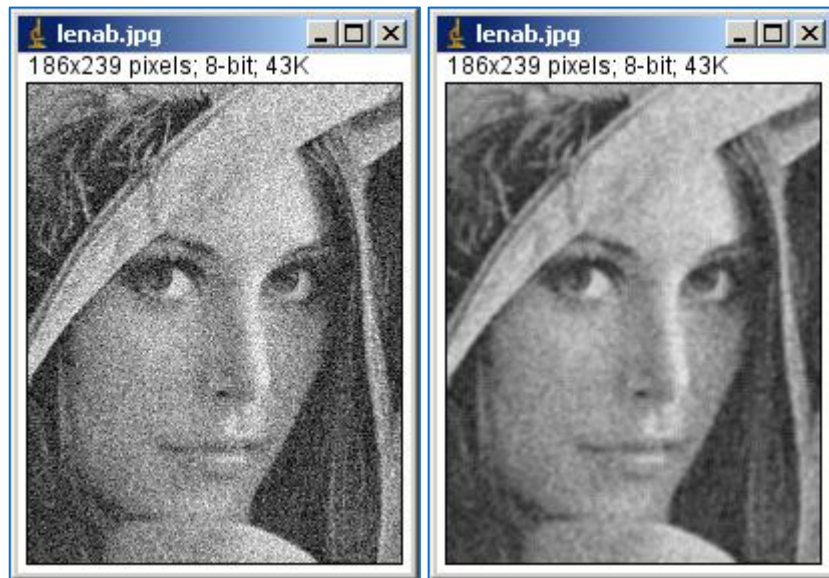


Figura 3.1.3. Izquierda: detalle de la imagen lena.jpg en escala de grises y con ruido gaussiano de media 0 y desviación estándar 25; derecha, la misma imagen después de ser procesada con la herramienta *Smooth*

En los apartados siguientes se describen otras herramientas que también pueden aplicarse en la eliminación de ruido y producen resultados de mayor calidad.

3.2. Sharpen

En este caso se agudiza (lo cual puede entenderse como el proceso inverso al anterior) la imagen activa con un filtrado espacial agudizador prefijado: se emplea un filtro de perfilado de bordes con la siguiente máscara 3x3:

Tabla 3.2.1. Máscara del filtro agudizador de perfilado de bordes que implementa la herramienta *Sharpen*

-1	-1	-1
-1	12	-1
-1	-1	-1

Al contrario que en el caso anterior, este filtro acentúa las altas frecuencias espaciales de la imagen, es decir, aquellos elementos que suponen cambios bruscos en el nivel de brillo: los bordes, los detalles finos, etc. Hay que tener en cuenta que también acentúa el ruido. En la Figura 3.2.1 se muestra un ejemplo de sus efectos:



Figura 3.2.1. Detalle de lena.jpg. De izquierda a derecha: original, tras aplicar *Sharpen* una vez y tras aplicar *Sharpen* dos veces

3.3. Find Edges

Al pulsar en *Find Edges* se aplica un detector de bordes de Sobel sobre la imagen activa. Este detector es otro filtro espacial, que en este caso consiste en las dos plantillas que se muestran en la Tabla 3.3.1, y que sirven para llevar a cabo la implementación discreta del operador gradiente aplicado al nivel de brillo de una imagen digital, el cual se entiende como una función discreta bidimensional (de ahí las dos plantillas).

Tabla 3.3.1. Máscaras del filtro detector de bordes de Sobel

1	2	1	-1	0	-1
0	0	0	2	0	-2
-1	-2	-1	1	0	-1

Con los dos valores que arrojan para cada píxel las plantillas se calcula la magnitud del gradiente del nivel de brillo en dicho punto haciendo la raíz cuadrada de sus cuadrados. Con todos los valores de la magnitud del gradiente se construye la imagen resultante, que será una imagen con un gran nivel de brillo en las zonas en las que en la imagen original había un gran gradiente en el nivel de brillo, es decir, en las que zonas en las que había cambios bruscos en el nivel de brillo, que suelen corresponder a los bordes (de ahí su nombre). Por el contrario, las zonas que en la imagen original eran uniformes, en la imagen resultante apenas tendrán nivel de brillo. En la Figura 3.3.1 se muestra el resultado de aplicar este filtro:



Figura 3.3.1. Izquierda: detalle de lena.jpg. Derecha: bordes de la imagen anterior obtenidos mediante *Find Edges*

3.4. Find Maxima

La herramienta *Find Maxima* encuentra máximos locales en el nivel de brillo de la selección hecha sobre la imagen activa (o, en su defecto, sobre toda la imagen). A partir de los máximos encontrados genera una máscara binaria de uno u otro tipo, o despliega la tabla de resultados con una u otra información en función de lo que indique el usuario en la interfaz que se despliega, la cual se muestra en la Figura 3.4.1.

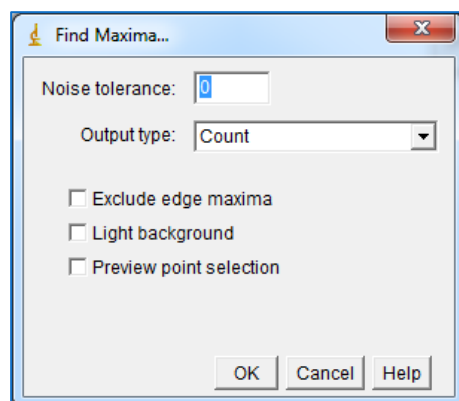


Figura 3.4.1. Interfaz *Find Maxima*

La interfaz de la herramienta *Find Maxima* contiene los siguientes campos:

- *Noise tolerance*. Permite determinar la cantidad de nivel de brillo en la que el máximo tiene que superar a los píxeles que le rodean para ser considerado como tal.
- *Output type*. Lista desplegable que permite elegir la salida que genera *Find Maxima*.
 - *Single Points*. Se genera una máscara binaria con un pixel negro en cada uno de los puntos en los que había un máximo local en la imagen original.

- *Maxima Within Tolerance.* Se genera una máscara binaria como la anterior, con un píxel negro en cada uno de los puntos en los que había un máximo local en la imagen original, pero además, para el entorno⁵ de cada máximo, por cada píxel cuyo brillo está por encima del umbral definido por *Noise tolerance* en la imagen original se añade un píxel negro en la máscara.
- *Segmented Particles.* Genera una máscara binaria con el resultado de segmentar la imagen original a partir de los máximos y según un algoritmo *Watershed* aplicado al nivel de brillo de la imagen.
- *Point Selection.* Genera sobre la imagen original una selección multipunto con un punto en cada máximo local.
- *List.* Abre la tabla de resultados y lista las coordenadas de cada máximo local
- *Count.* Abre la tabla de resultados y muestra el total de máximos locales procesados.
- *Exclude edge máxima.* Marcando esta casilla se excluyen los máximos si los píxeles de su entorno que están por encima del umbral tocan el borde de la imagen.
- *Light background.* Marcando esta casilla se considera que el fondo es más brillante que los objetos de interés.
- *Preview point selection.* Marcando esta casilla se muestra una vista previa de los máximos locales a través de una selección multipunto con un punto sobre cada uno de ellos.

En la Figura 3.4.2 se muestran ejemplos de las diferentes salidas que puede generar *Find Maxima*:

⁵ Para más detalles acerca de la definición del entorno de cada máximo local, consultar la siguiente dirección: <http://imagej.1557.x6.nabble.com/Find-maxima-algorithm-discription-td3683914.html>

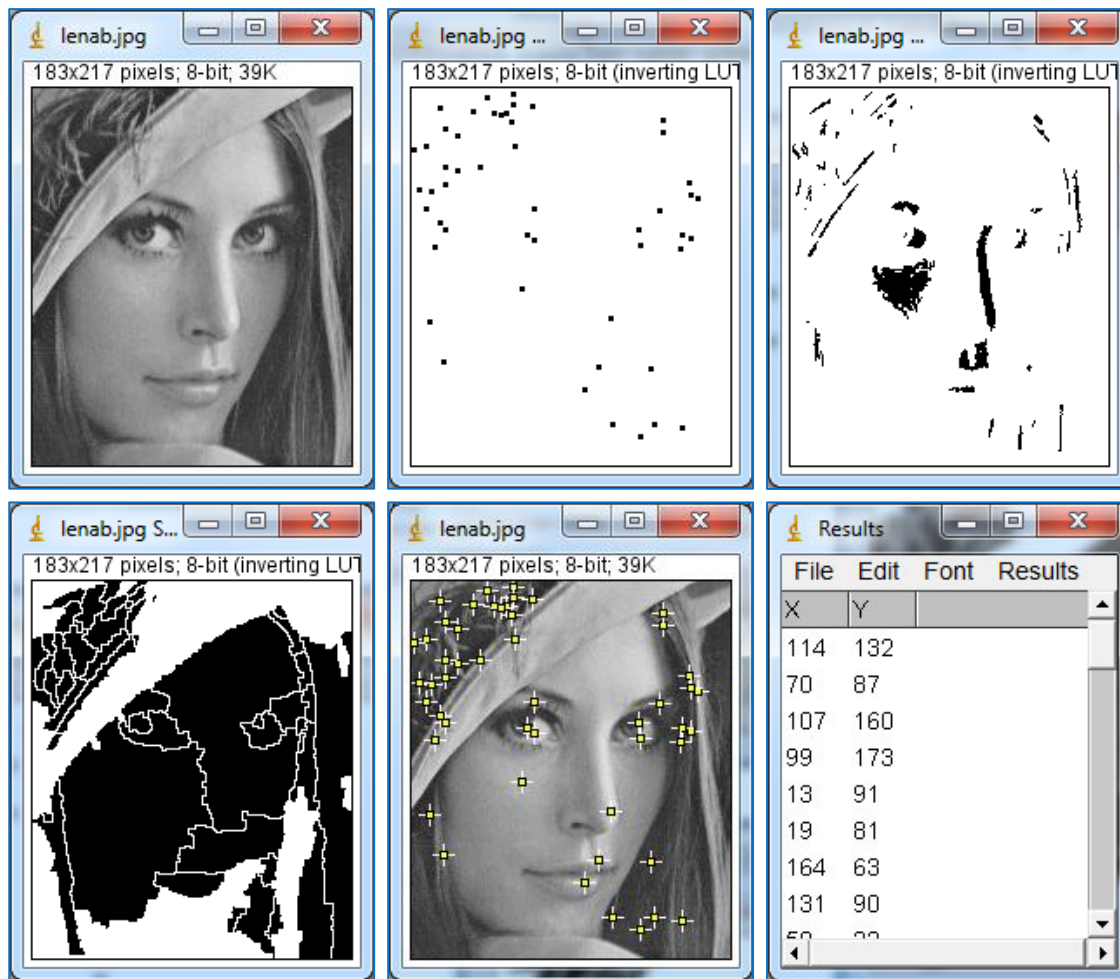


Figura 3.4.2. De izquierda a derecha y de arriba a abajo: imagen original, máscara *Single Points* correspondiente, máscara *Maxima Within Tolerance*, máscara *Segmented Particles*, selección multipunto correspondiente y tabla de resultados correspondiente, todos ellos generados con *Find Maxima* (con *Noise tolerance* de 25)

3.5. Enhance Contrast

Esta herramienta permite optimizar el contraste de la imagen mediante diferentes técnicas. Para especificar la técnica que se prefiere usar y también para determinar otros parámetros de la operación se presenta la interfaz que se presenta en la Figura 3.5.1:

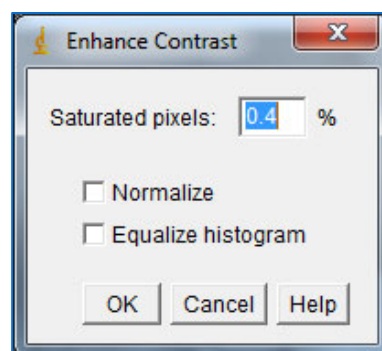


Figura 3.5.1. Interfaz *Enhance Contrast*

Con el campo *Saturated pixels* se permite especificar el porcentaje de píxeles saturados que dejará la operación. Evidentemente, cuanto mayor sea este valor, mayor será el contraste global de la imagen. Marcando la casilla *Normalize*, se optimiza el contraste de la imagen a base de recalcular el valor de los píxeles mediante una técnica de *stretching* (ver la herramienta *Brightness/Contrast* en el apartado 2.2 Submenú *Adjust*), de modo que el rango dinámico de la imagen procesada coincida con el rango dinámico máximo posible para cada tipo de imagen (256 para imágenes de tipo *8-bit*, 65536 para imágenes de tipo *16-bit*,...). Por último, marcando la casilla *Equalize histogram* se optimiza el contraste de la imagen activa ecualizando su histograma (o, en caso de que existiera una selección sobre ella, en base al histograma de la selección). En la Figura 3.5.2 se muestran ejemplos de optimización de contraste con la herramienta *Enhance Contrast*:



Figura 3.5.2. De izquierda a derecha: imagen original, imagen con contraste optimizado a través de *stretching* (0,5% de píxeles saturados) e imagen con contraste optimizado a través de ecualización del histograma (0,5% de píxeles saturados)

3.6. Submenú Noise

Bajo este epígrafe se agrupan varias herramientas destinadas a contaminar la imagen activa con ruidos de diferentes características y también a restaurar imágenes contaminadas con diferentes tipos de ruido:

- *Add Noise*. Añade a la imagen activa o selección un ruido gaussiano prefijado de media 0 y desviación estándar 25. Un ejemplo de esto se muestra en la Figura 3.6.1:

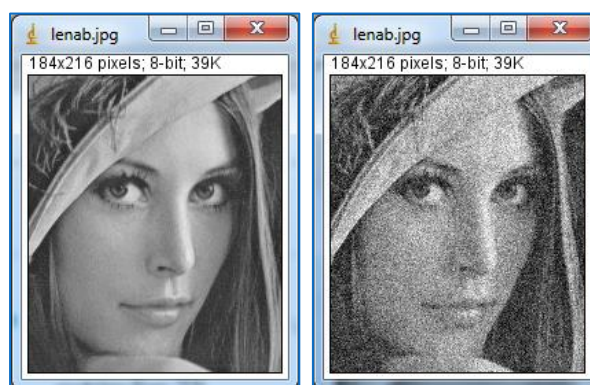


Figura 3.6.1. Izquierda: imagen original. Derecha: imagen anterior contaminada con ruido mediante *Add Noise*

- *Add Specified Noise*. Añade a la imagen activa o selección un ruido gaussiano de media 0 y de la desviación estándar especificada por el usuario en el diálogo que se despliega. En la Figura 3.6.2 se muestra una imagen contaminada de esta forma:

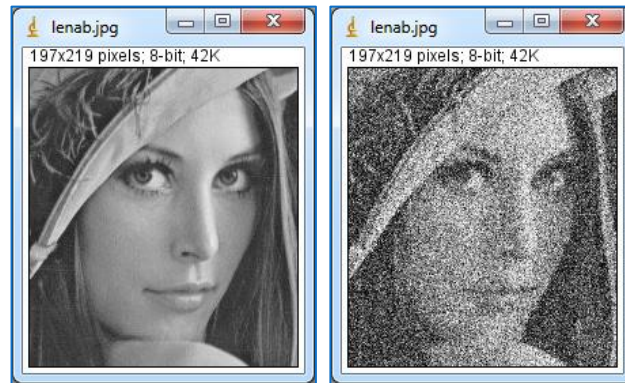


Figura 3.6.2. Izquierda: imagen original. Derecha: imagen anterior contaminada con ruido mediante *Add Specified Noise* (con desviación estándar de valor 50)

- *Salt and Pepper*. Añade a la imagen activa o selección un ruido de tipo *Salt and Pepper* (o Sal y Pimienta) prefijado: el porcentaje de píxeles afectados por el ruido es 5% (el 2,5% pasa a ser negros y el otro 2,5% pasa a ser blanco). En la Figura 3.6.3 se muestra su efecto:



Figura 3.6.3. Izquierda: imagen original. Derecha: imagen anterior contaminada con ruido de tipo *Salt and Pepper* mediante la herramienta *Salt and Pepper*

- *Depeckle*. Se trata de una herramienta de restauración. Es un filtro espacial de mediana de dimensiones 3x3 (es decir, cada pixel es sustituido por la mediana del conjunto formado por los 9 píxeles que abarca una matriz 3x3 centrada en él). Es especialmente efectivo a la hora de restaurar ruido de tipo *Salt and Pepper*, aunque suaviza ligeramente la imagen. A continuación, en la Figura 3.6.4, se muestra un ejemplo de su aplicación:



Figura 3.6.4. Izquierda: imagen original. Centro: imagen contaminada con ruido de tipo *Salt and Pepper*. Derecha: imagen anterior restaurada usando *Depeckle*

- Remove Outliers*. Se trata de nuevo de una herramienta de restauración. Es un filtro espacial *Outlier* que el usuario puede configurar mediante un menú emergente. Este tipo de filtro actúa sobre aquellos píxeles cuyo nivel de brillo se desvía de la mediana del nivel de brillo de su entorno en un valor superior a un umbral prefijado; el nivel de brillo de los píxeles señalados se sustituye por dicha mediana. En el menú emergente el usuario puede especificar el radio en píxeles del área circular que se considera como entorno (*Radius*), el umbral mencionado (*Threshold*) y si se consideran los píxeles que se desvían por encima del umbral (*Bright*) o por debajo (*Dark*). También permite mostrar una vista previa del resultado (marcando la casilla *Preview*). Este tipo de filtro es, como el filtro espacial de mediana, efectivo en imágenes contaminadas con ruido de tipo *Salt and Pepper*, pero además presenta la ventaja de conservar la nitidez de la imagen. El filtro de mediana actúa sobre los píxeles contaminados, pero también sobre los demás, lo que implica que además de recalcular el valor de los píxeles contaminados recalcula el valor del resto de los píxeles, que debería conservarse. Como la operación que lleva a cabo para cada píxel es reemplazar su nivel de brillo por la mediana de los niveles de brillo de su entorno, termina por emborronar la imagen. En este sentido, el filtro *Outlier* puede entenderse como un filtro de mediana selectivo: configurando el umbral adecuadamente, el filtro sólo afectará a los píxeles contaminados, que generalmente tendrán un nivel de brillo muy desviado de la mediana de niveles de brillo de su entorno, y conservará intactos los demás píxeles, eliminando el ruido y manteniendo la nitidez de la imagen. En la se muestra una comparativa de restauraciones con *Depeckle* y con *Remove Outliers*:

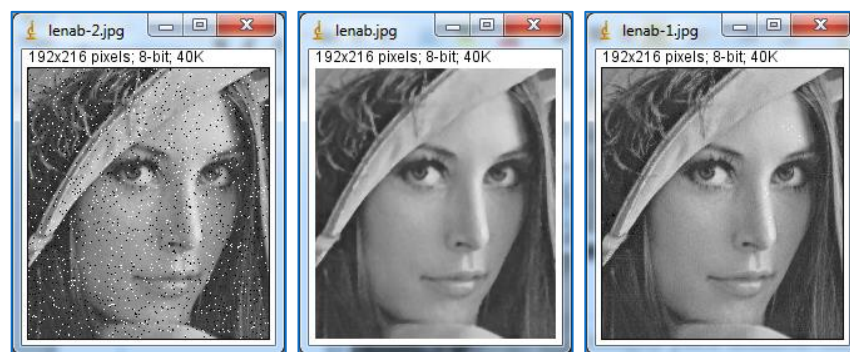


Figura 3.6.5. De izquierda a derecha: original, restaurada con *Depeckle* y restaurada con *Remove Outliers*

- *Remove NaNs*. Esta última herramienta también está orientada a la restauración de imágenes. Se trata de un filtro espacial que elimina los *NaN* (*Not a Number* o No es un Número) que a veces aparecen en imágenes de tipo *32-bit* y los reemplaza por la mediana de los niveles de brillo de los píxeles de su entorno. Una interfaz permite al usuario definir el radio en píxeles del área circular que se considera entorno. Además, la casilla *Preview* facilita la vista previa de la operación.

3.7. Submenú Shadows

Esta herramienta permite filtrar la imagen activa con filtros espaciales orientados a crear sombras en la imagen, simulando que la imagen estuviera iluminada desde diferentes localizaciones que el usuario puede elegir: norte, noreste, este, sudeste, sur, sudoeste, oeste y noroeste (*North, Northeast, East, Southeast, South, Southwest, West* y *Northwest* respectivamente). En la Tabla 3.7.1 se muestran las plantillas empleadas para implementar algunos de estos filtros:

Tabla 3.7.1. Plantillas correspondientes a los filtros norte, sur, este y oeste respectivamente

1	2	1	-1	-2	-1	-1	0	-1	1	0	-1
0	1	0	0	1	0	-2	1	-2	2	1	-2
-1	-2	-1	1	2	1	-1	0	-1	1	0	-1

En el caso del filtro norte, por ejemplo, se puede deducir de su plantilla que al valor de cada píxel (corresponde al 1 en el centro de la plantilla) se le añade un incremento de brillo si hay cambios en el nivel de brillo en su entorno en sentido vertical (debido a que la fila superior y la inferior son opuestas), y que dicho incremento es mayor cuanto mayores son los cambios. Además, este incremento es positivo (el píxel se hace más brillante) si se pasa de una zona brillante a una oscura y negativo (el píxel se hace más oscuro) si se pasa de una zona oscura a una zona brillante. Es decir, se abrillanta la parte superior de los bordes horizontales y se oscurece su parte inferior, tal y como sucedería si se iluminase la imagen desde arriba (desde el norte). En la Figura 3.7.1 se muestran varios ejemplos de los resultados que se consiguen con esta herramienta:

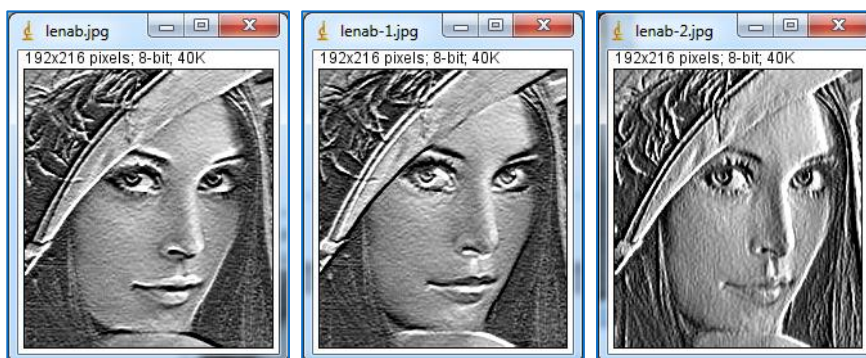


Figura 3.7.1. Izquierda: lena.jpg filtrada con *Shadows-North*. Centro: lena.jpg filtrada con *Shadows-South*. Derecha: lena.jpg filtrada con *Shadows-West*

3.8. Submenú Binary

En este submenú se agrupan las herramientas aplicables a imágenes binarias. En este tipo de imágenes los píxeles solo pueden tomar dos valores, 0 o 1, aunque a la hora de representarlas se traducen en 0 (negro) y 255 (blanco). A continuación se revisan las herramientas presentes en este submenú:

- *Make Binary*. Esta opción convierte la imagen activa en una imagen binaria. Para determinar qué píxeles pasarán a valer 0 y qué píxeles pasarán a valer 1 es necesario establecer un umbral: en la imagen original, a los píxeles cuyo nivel de brillo esté por debajo del umbral se les asignará 0, y a los demás se les asignará 1 (o al revés). Si previamente se ha definido un umbral con la herramienta *Threshold* (submenú *Adjust*, menú *Image*), se toma dicho umbral. De no ser así, el umbral se calcula de forma automática en base al histograma de la imagen (o selección, en caso de existir). Antes de llevar a cabo la operación se muestra un menú que permite al usuario elegir a qué píxeles se les asigna el negro y a qué píxeles se les asigna el blanco. En la Figura 3.8.1 se muestra un ejemplo de la aplicación de esta herramienta.



Figura 3.8.1. Izquierda: imagen original. Derecha: imagen binarizada automáticamente

- *Convert to Mask*. Implementa la misma funcionalidad que la herramienta anterior, *Make Binary*.
- *Erode*. Esta herramienta erosiona los objetos de las imágenes binarias: hace que los píxeles más externos de su contorno pasen a formar parte del fondo. En la Figura 3.8.2 se muestra un ejemplo:

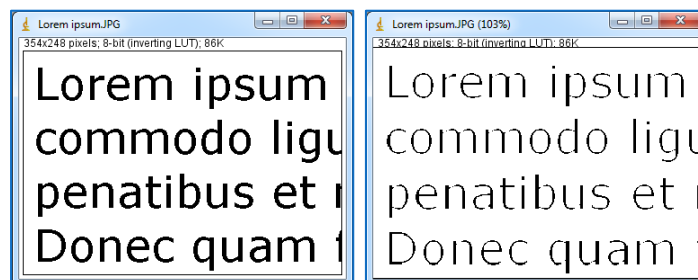


Figura 3.8.2. Izquierda: imagen binaria. Derecha: imagen anterior erosionada con *Erode*

- *Dilate*. Al contrario que la herramienta anterior, *Dilate* añade píxeles al contorno de los objetos (los dilata). En la Figura 3.8.3 se muestra un ejemplo:

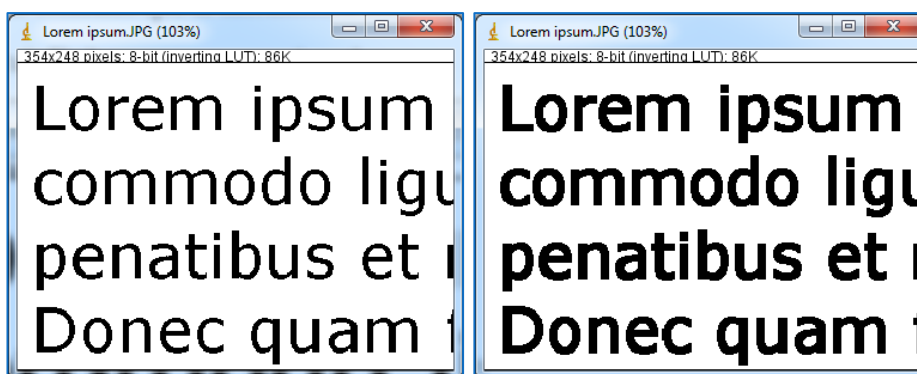


Figura 3.8.3. Izquierda: imagen binaria. Derecha: imagen anterior dilatada con *Dilate*

- *Open*. Aplica la herramienta *Erode* y después la herramienta *Dilate*. En la imagen resultante desaparecen los píxeles aislados, que son eliminados por completo tras aplicar *Erode*. A los objetos más gruesos también les sucede esto, pero por ser más gruesos no desaparecen por completo, y vuelven a recuperar su forma cuando se aplica *Dilate*. En la Figura 3.8.4 se muestra un ejemplo de esto:

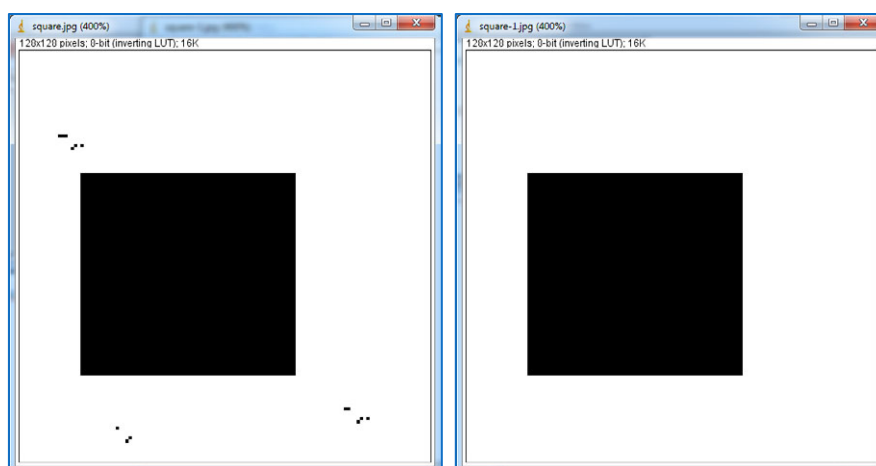


Figura 3.8.4. Izquierda: imagen binaria. Derecha: imagen anterior tras aplicar *Open*

- *Close*. Aplica la herramienta *Dilate* y después la herramienta *Erode*. En la imagen resultante, los pequeños huecos que los objetos podrían contener aparecen rellenos. Esto se debe a que la primera operación, *Dilate*, dilata los objetos también por sus contornos interiores, de forma que si los huecos son lo suficientemente pequeños terminan por desaparecer completamente. Al aplicar la siguiente operación, *Erode*, los huecos no reaparecen porque no hay contornos interiores que erosionar. Para el resto de objetos, la erosión contrarresta la dilatación inicial, de modo que mantienen su forma. En la Figura 3.8.5 se muestra un ejemplo:

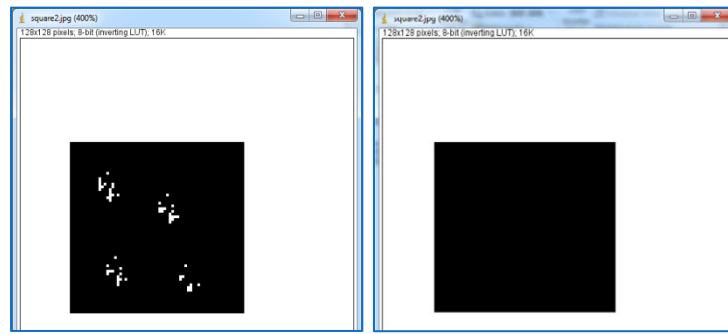


Figura 3.8.5. Izquierda: imagen binaria. Derecha: imagen anterior tras aplicar *Close*

- *Outline*. Esta herramienta delinea los objetos en las imágenes binarias, es decir, deja su contorno (con un píxel de anchura) y elimina todo su interior, que pasa a formar parte del fondo.

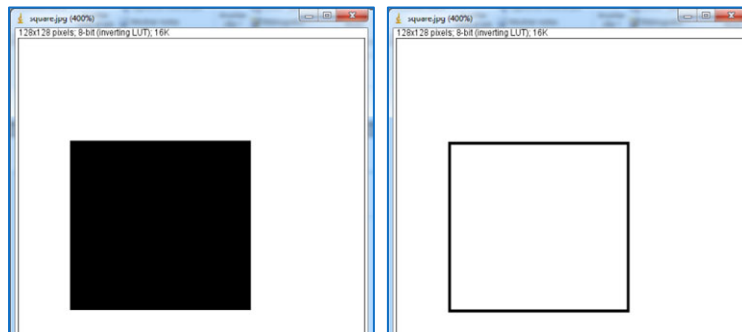


Figura 3.8.6. Izquierda: imagen binaria. Derecha: imagen anterior tras aplicar *Outline*

- *Fill Holes*. Como se ha comentado, la herramienta *Close* es capaz de eliminar pequeños huecos dentro de los objetos de imágenes binarias. Sin embargo, si estos huecos son demasiado grandes, no es posible eliminarlos por completo de esta forma. La herramienta *Fill Holes* permite rellenar los huecos en los objetos de forma más efectiva siguiendo una filosofía diferente a la que sigue *Close*: se entiende que los grupos de píxeles con conectividad 4 (aquellos píxeles vecinos que tienen en contacto directo sus bordes) que pertenecen al fondo y a la vez están rodeados por objetos son huecos. Una vez detectados los huecos, sencillamente se rellenan con negro, con lo que pasan a formar parte del objeto. En la Figura 3.8.7 se muestra una comparativa con los resultados que se obtienen empleando *Close* y *Fill Holes*:

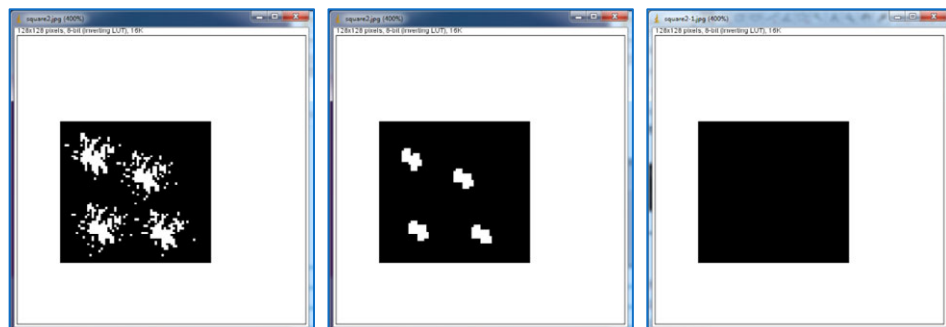


Figura 3.8.7. Izquierda: imagen binaria original con un objeto con huecos. Centro: resultado de aplicar a la imagen anterior la herramienta *Close*. Derecha: resultado de aplicar a la imagen anterior la herramienta *Fill Holes*.

- *Skeletonize*. Esta herramienta erosiona los objetos de las imágenes binarias hasta que sólo tienen un píxel de ancho (deja su esqueleto). En la Figura 3.8.8 se muestra un ejemplo de su efecto:

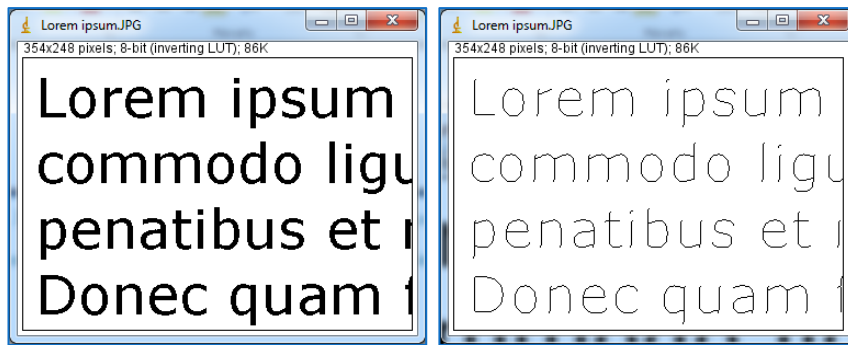


Figura 3.8.8. Izquierda: imagen binaria. Derecha: imagen anterior tras aplicar *Skeletonize*

- *Distance Map*. Esta herramienta genera mapas de distancias. Los mapas de distancias son imágenes no binarias que se crean a partir de imágenes binarias de forma que cada píxel del mapa tiene un nivel de brillo equivalente a la distancia que había desde el píxel que ocupaba la misma posición en la imagen binaria hasta la parte del fondo más cercana. Por tanto, los píxeles del fondo tendrán un nivel 0 y los píxeles más interiores de los objetos tendrán un nivel muy alto. Hay que recordar que la *LUT* está invertida, y por tanto los píxeles del fondo, con nivel 0, serán blancos, y los píxeles interiores de los objetos, con niveles altos, serán oscuros. Además, en función del tipo de imagen de salida (que se elige en *Options*, en este mismo submenú), puede ocurrir que no haya suficientes niveles de gris para describir todas las distancias. Por ejemplo, no sería extraño que los píxeles internos de un objeto binario distaran de la parte del fondo más cercana más de 255 píxeles, así que sus distancias no se podrían describir fielmente en mapas de distancias de tipo *8-bit*, en las que sólo hay 256 niveles de brillo posibles, desde el 0 hasta el 255. En estos casos, sencillamente se les asignaría el nivel máximo que se puede describir en la imagen, lo que se puede entender como saturar el mapa de distancias. En la Figura 3.8.9 se muestra una imagen binaria junto a su correspondiente mapa de distancias:

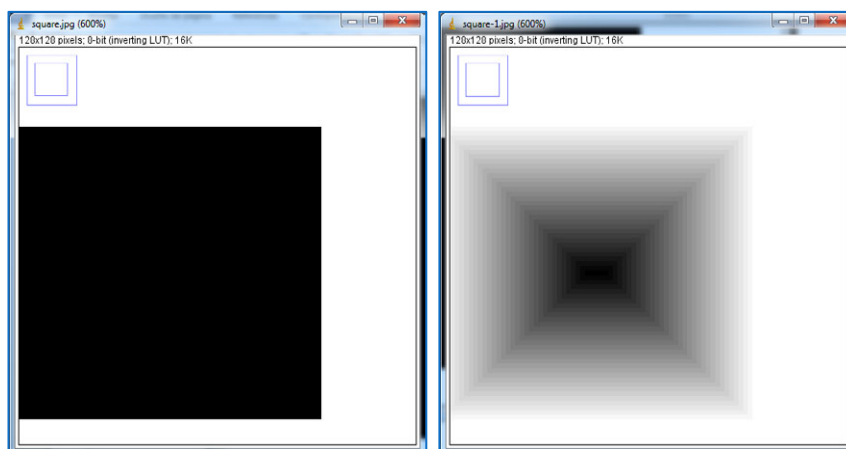


Figura 3.8.9. Izquierda: imagen binaria. Derecha: mapa de distancias de la imagen anterior.

- *Ultimate Points*. Esta herramienta genera a partir de una imagen binaria otra imagen no binaria con los *Ultimates Points* correspondientes a los objetos de la primera. Los *Ultimates Points* serían los máximos locales del mapa de distancias de la correspondiente imagen binaria. También se les conoce como UEPs (*Ultimate Eroded Points* o Últimos Puntos Erosionados) porque serían los últimos puntos en desaparecer en caso de que se erosionaran repetidamente los objetos de la imagen binaria. En la Figura 3.8.10 se muestra una imagen binaria junto a sus *Ultimate Points*.

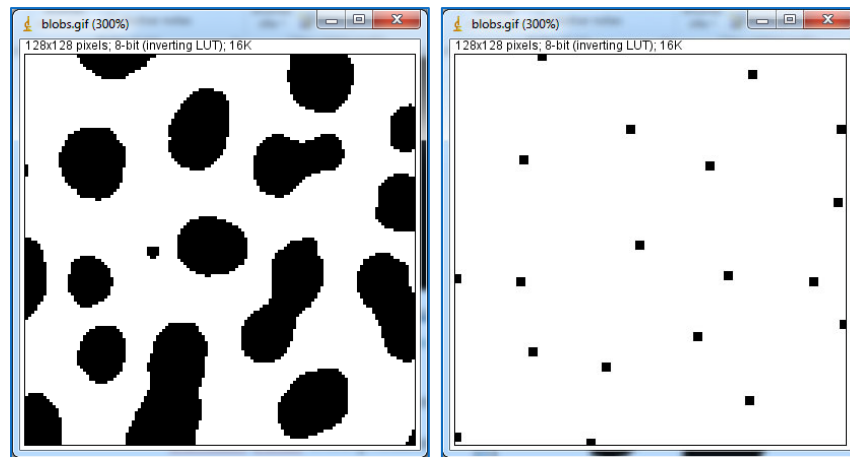


Figura 3.8.10. Izquierda: imagen binaria. Derecha: *Ultimate Points* de la imagen anterior

- *Watershed*. Esta herramienta permite separar objetos de imágenes binarias que en realidad corresponden a dos o más partículas que están en contacto. Para ello se emplea una segmentación de tipo *Watershed* que, esencialmente, se basa en calcular los *Ultimate Points* de la imagen binaria y dilatarlos repetidamente hasta que se alcance el borde del objeto, que es lo que sucederá en el caso de los objetos que corresponden a partículas individuales. Sin embargo, según este algoritmo, en los objetos correspondientes a dos o más partículas en contacto se encontrarán dos o más *Ultimate Points* (es decir, dos o más máximos locales en el mapa de distancias), y al dilatarlos interferirán unos con otros antes de alcanzar el borde del objeto original. En base a la fase de la dilatación en la que interfieran se generarán los nuevos objetos a partir del objeto original. Esto se muestra en la Figura 3.8.11:



Figura 3.8.11. Izquierda: imagen binaria. Derecha: imagen anterior con objetos segmentados usando *Watershed*

- *Voronoi*. Esta herramienta genera imágenes no binarias resultantes de segmentar las imágenes binarias con líneas que atraviesan el fondo por los puntos equidistantes a los dos objetos más cercanos. Como la nueva imagen es no binaria, se pueden utilizar diferentes niveles de brillo, lo cual se aprovecha en las líneas que segmentan la imagen: los puntos de las líneas lejanos a los dos objetos más cercanos tienen mayor nivel (como la *LUT* está invertida, significa que son más oscuros) y los puntos cercanos a los objetos que los rodean tienen menor nivel (son más claros). El nivel de brillo de estos puntos tiene el mismo valor que su distancia en píxeles a los dos objetos que los rodean. Un ejemplo de esto se muestra en la Figura 3.8.12.

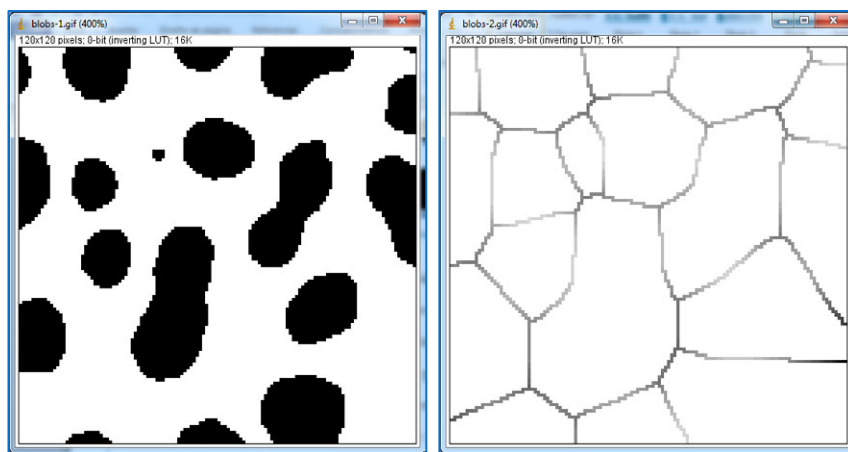


Figura 3.8.12. Izquierda: imagen binaria. Derecha: imagen anterior tras aplicar la segmentación *Voronoi*

- *Options*. Despliega un menú en el que el usuario puede configurar algunas características globales de las operaciones binarias:
 - *Iterations*. En este campo de texto se puede definir de forma manual el número de iteraciones que se llevan a cabo para las operaciones de erosión (*Erode*), dilatación (*Dilate*), apertura (*Open*) y cierre (*Close*).
 - *Count*. Permite determinar el número mínimo de píxeles del fondo adyacentes a un píxel del borde de un objeto que son necesarios para eliminar dicho píxel cuando se lleva a cabo una erosión. Es decir, si en este campo se introduce, por ejemplo, 5 y en la imagen hay un objeto cuyo contorno tiene una parte que tan sólo dista 1 píxel de otro objeto (entre medias sólo hay un píxel de fondo), los píxeles de dicha parte del contorno no se eliminaría si se erosionara la imagen. El valor introducido también se toma como el número mínimo de píxeles del objeto adyacentes a un píxel del fondo que son necesarios para convertir dicho píxel en parte del objeto cuando se lleva a cabo una dilatación.
 - *Black background*. Como se puede comprobar tras leer los apartados anteriores, la mayoría de las operaciones binarias consideran que las imágenes binarias contienen elementos de dos tipos: objetos y fondo. Por defecto se invierte la *LUT*, es decir, se considera que los píxeles negros forman parte de objetos (tienen valor 1 o 255) y que los píxeles blancos forman parte del fondo (tienen valor 0). Sin

embargo, esta consideración se puede invertir marcando la casilla *Black Background* (se reinvierte la *LUT*).

- *Pad edges when eroding*. Por defecto, al erosionar con la herramienta *Erode* (e intrínsecamente con la herramienta *Close*), se erosionan los objetos por todo su contorno, incluso por la parte del contorno que está en contacto directo con el borde de la imagen. Para evitar esto último, se puede marcar esta casilla.
- *EDM output*. Esta lista desplegable permite elegir el tipo de la imagen que generan las herramientas que producen imágenes no binarias (*Distance Map*, *Ultimate Points* y *Voronoi*): *8-bit*, *16-bit* y *32-bit*.
- *Do*. Permite elegir entre casi todas las operaciones binarias para obtener una vista previa de sus efectos sobre la imagen activa.

3.9. Submenú Math

En este submenú se agrupan varias operaciones matemáticas sencillas que se aplican a las imágenes píxel a píxel de modo que el nivel de brillo de cada píxel de la imagen resultante sólo depende del nivel de brillo que tenía el mismo píxel en la imagen original (por lo que se conocen como operaciones de punto) [8].

- *Add*. Se suma al nivel de brillo de cada píxel de la imagen activa la constante (positiva o negativa) especificada por el usuario. Evidentemente, un valor positivo aclarará la imagen y un valor negativo la hará más oscura. Si el valor es demasiado positivo o demasiado negativo puede que el nivel de brillo de algunos píxeles sobrepase por encima o por debajo los extremos del rango dinámico de la imagen; en tales casos, se asigna a dichos píxeles los valores máximos o mínimos asignables en función del tipo de la imagen.
- *Subtract*. Se resta al nivel de brillo de cada píxel de la imagen activa la constante especificada por el usuario, que de nuevo puede ser positiva o negativa. Se pueden hacer las mismas consideraciones que se hicieron en el caso anterior.
- *Multiply*. Se multiplica el nivel de brillo de cada píxel de la imagen activa por la constante especificada por el usuario, que también puede ser positiva o negativa. Sin embargo, como no puede haber niveles de brillo negativos, a cualquiera de estos niveles se les asigna el nivel 0, por lo que una constante negativa hará que la imagen se vuelva completamente negra. Por otro lado, al contrario que en el caso de la suma, una multiplicación por un valor superior a 1 no aclarará todos los píxeles por igual, sino que aquellos píxeles que ya eran brillantes originalmente aumentarán su brillo más que los píxeles oscuros. Por ello, a no ser que se usen valores muy cercanos a 1, no es extraño quemar la imagen, es decir, saturarla excediendo su margen dinámico por su extremo superior. Evidentemente, valores positivos inferiores a 1 oscurecerán la imagen (y oscurecerán más los píxeles que eran brillantes que los píxeles que eran oscuros).

- *Divide*. Se divide el nivel de brillo de cada píxel de la imagen activa por la constante especificada por el usuario. Una vez más, si se especifica un valor negativo, la imagen pasará a ser totalmente negra. Por lo demás, se pueden hacer las consideraciones inversas a las hechas en el caso de *Multiply*: valores superiores a 1 oscurecerán la imagen (y más las partes más brillantes) y valores positivos inferiores a 1 la aclararán (y más en las partes más claras).
- *AND*. Se lleva a cabo la operación bit a bit AND con el valor binario del nivel de brillo de cada píxel de la imagen activa y la secuencia binaria especificada por el usuario. Por ejemplo, para imágenes de tipo *8-bit*, la secuencia 1111 1111 las deja como están y la secuencia 0000 0000 las vuelve completamente negras.
- *OR*. Se lleva a cabo la operación bit a bit OR con el valor binario del nivel de brillo de cada píxel de la imagen activa y la secuencia binaria especificada por el usuario. Por ejemplo, para imágenes de tipo *8-bit*, la secuencia 1111 1111 las vuelve completamente blancas y la secuencia 0000 0000 las deja como están.
- *XOR*. Se lleva a cabo la operación bit a bit XOR con el valor binario del nivel de brillo de cada píxel de la imagen activa y la secuencia binaria especificada por el usuario. Por ejemplo, para imágenes de tipo *8-bit*, la secuencia 1111 1111 las invierte y la secuencia 0000 0000 las deja como están.
- *Min*. Para todos los píxeles de la imagen activa, cualquier nivel de brillo inferior al valor especificado por el usuario se sustituye por dicho valor, de forma que el nivel de brillo mínimo que se puede encontrar en la imagen activa es el indicado por el usuario.
- *Max*. Para todos los píxeles de la imagen activa, cualquier nivel de brillo superior al valor especificado por el usuario se sustituye por dicho valor, de forma que el nivel de brillo máximo que se puede encontrar en la imagen activa es el indicado por el usuario.
- *Gamma*. Aplica la corrección gamma a la imagen activa, permitiendo al usuario especificar el parámetro γ . La ecuación correspondiente a la corrección gamma se muestra en la expresión (3.9.2)

$$v = \left(\frac{u}{255}\right)^{\gamma} \quad 255, 0,1 \leq \gamma \leq 5,0 \quad (3.9.2)$$

Valores de γ superiores a 1 oscurecen la imagen y valores inferiores la aclaran según curvas como las de la Figura 3.9.1. Para las imágenes de tipo *16-bit* se modifica la expresión (3.9.2): en vez escalarse el resultado a 255 como en este caso (en el que se consideran imágenes de tipo *8-bit*), se escala según el rango dinámico de la imagen (la diferencia entre sus niveles de brillo máximo y mínimo). En las imágenes de tipo *RGB* se aplica la corrección a cada canal por separado.

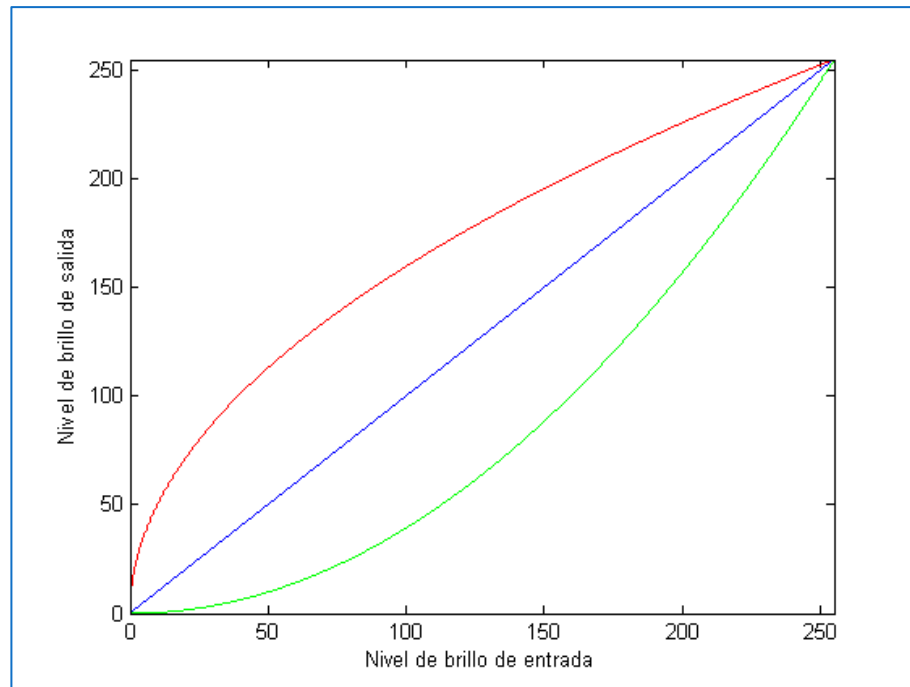


Figura 3.9.1. Curvas de corrección gamma en un gráfico que enfrenta brillo de salida a brillo de entrada para una imagen de tipo 8-bit (es decir, de 0 a 255). Rojo: $\gamma=0,5$. Azul: $\gamma=1$. Verde: $\gamma=2$

- *Set*. Asigna a todos los píxeles que abarque la selección (o, en su defecto, a todos los de la imagen activa) el nivel de brillo elegido por el usuario.
- *Log*. Aplica al nivel de brillo de cada píxel de la imagen activa la expresión (3.9.3):

$$v = \ln(u) \frac{255}{\ln(255)} \quad (3.9.3)$$

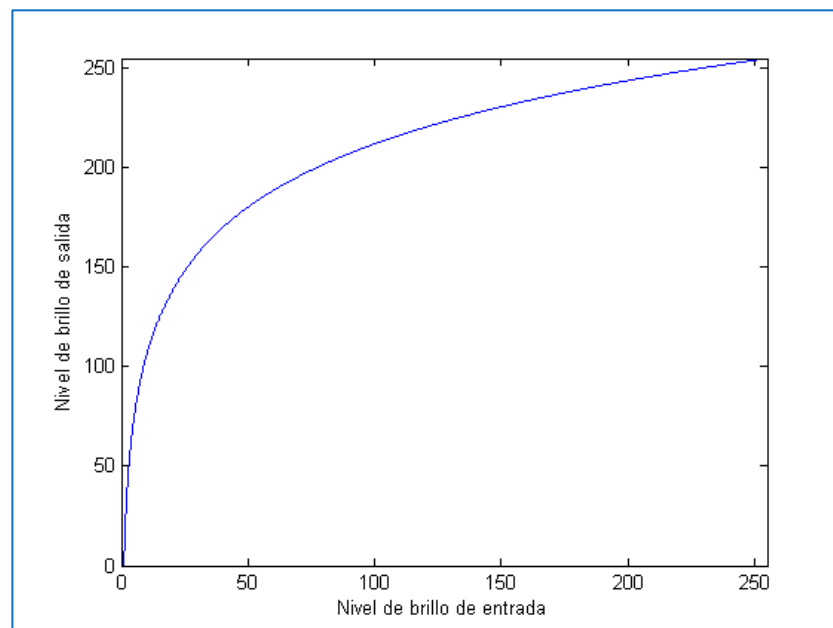


Figura 3.9.2. Función logarítmica correspondiente a la expresión (3.9.3) en un gráfico que enfrenta brillo de salida a brillo de entrada para una imagen de tipo 8-bit (es decir, de 0 a 255).

Con esta operación se consigue aumentar la cantidad de margen dinámico destinado a describir los niveles de brillo más bajos (a costa de describir los niveles de brillo altos con un menor margen dinámico), tal y como muestra la Figura 3.9.2.

Esto es útil para poder apreciar visualmente con mayor comodidad la información que contienen las imágenes muy oscuras (que se aclaran). Una vez más, para las imágenes de tipo *16-bit* se vuelve a escalar en función de su margen dinámico, y para las imágenes de tipo *RGB Color* se aplica por separado a cada canal.

- *Exp.* Aplica al nivel de brillo de cada píxel de la imagen activa la expresión (3.9.4):

$$v = 254^{\frac{u}{255}} \quad (3.9.4)$$

Al contrario que en el caso anterior, con esta operación se consigue aumentar la cantidad de margen dinámico destinado a describir los niveles de brillo más altos (a costa de describir los niveles de brillo bajos con un menor margen dinámico), tal y como muestra la Figura 3.9.3. Esto es útil para poder apreciar visualmente con mayor comodidad la información que contienen las imágenes muy brillantes (que se oscurecen).

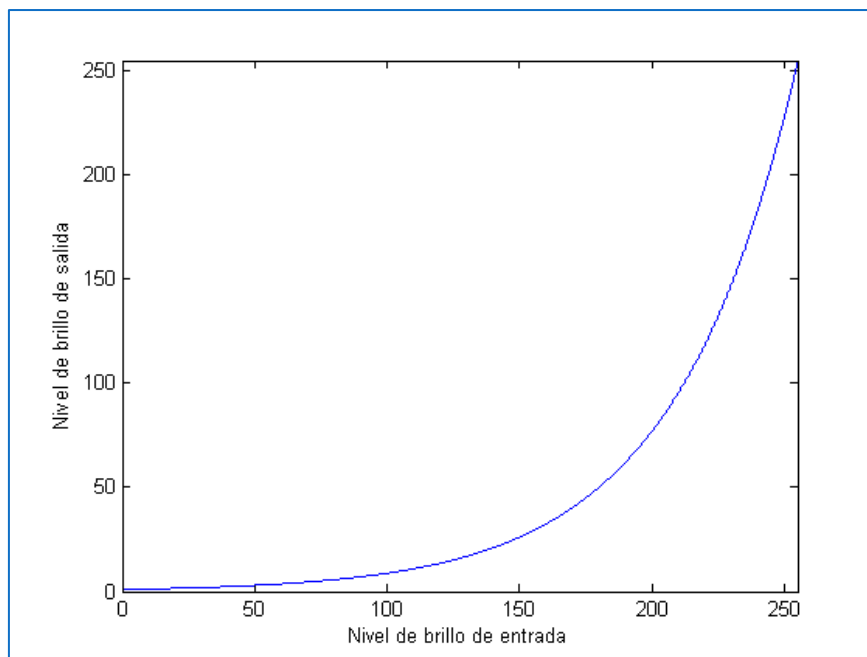


Figura 3.9.3. Función exponencial correspondiente a la expresión (3.9.4) en un gráfico que enfrenta brillo de salida a brillo de entrada para una imagen de tipo *8-bit* (es decir, de 0 a 255).

- *Square.* El nivel de brillo de cada píxel de la imagen activa se sustituye por su cuadrado o por el nivel de brillo máximo en caso de que su cuadrado sea superior a éste.
- *Square Root.* El nivel de brillo de cada píxel de la imagen activa se sustituye por su raíz cuadrada.

- *Reciprocal*. Sólo aplicable a imágenes de tipo *32-bit*, sustituye el nivel de brillo u de cada píxel de la imagen activa por su inverso o recíproco, $1/u$.
- *NaN Background*. Sólo aplicable a imágenes de tipo *32-bit*, sustituye los píxeles que no han sido previamente umbralizados por *NaN*.
- *Abs*. Sólo aplicable a imágenes de tipo *signed 16-bit* (tipo *16-bit* con valores enteros) y de tipo *32-bit*, reemplaza el nivel de brillo de cada píxel por el valor absoluto de dicho nivel.
- *Macro*. A través del menú emergente, permite definir manualmente y aplicar una operación aritmética compleja (una combinación de las operaciones sencillas revisadas hasta ahora) usando el lenguaje *macro* de ImageJ.

3.10. Submenú FFT

En este submenú se agrupan las herramientas de procesamiento relacionadas con el dominio de la frecuencia espacial.

- *FFT*. Esta herramienta hace uso de un algoritmo de tipo *FFT* bidimensional⁶ (*Fast Fourier Transform*, o Transformada Rápida de Fourier) para calcular la transformada de Fourier de la imagen activa. Esta transformada, que se almacena como una imagen de tipo *32-bit*, no se muestra; en su lugar, se muestra una imagen adjunta a la primera que puede contener diferentes tipos de información sobre el espectro de la imagen activa en función de lo que se indique en el apartado correspondiente del menú emergente *Options*, en este mismo submenú. Por defecto, la imagen que se muestra es de tipo *8-bit* y contiene la densidad espectral de potencia de la imagen activa, y aparece centrada (con la componente continua en el centro de la imagen) y procesada con una operación de tipo logaritmo para facilitar su visualización (de lo contrario sería demasiado oscura para distinguir los detalles más relevantes). En la Figura 3.10.1 se muestra un ejemplo de esto.

Por otro lado, a diferencia de lo que ocurre con el resto de imágenes, cuando el cursor se desplaza sobre las imágenes que contienen información del espectro de otras imágenes, en la barra de estado de ImageJ no se muestran coordenadas cartesianas, sino coordenadas polares: el ángulo, en grados, y el radio, en píxeles por ciclo (o en unidades por ciclo en caso de que la imagen original hubiera sido calibrada espacialmente de forma previa) del píxel apuntado por el cursor. Esto se muestra en la Figura 3.10.2. En concreto, en esta figura se muestran, destacadas en rojo, las coordenadas polares del píxel central de la imagen de densidad espectral de potencia que aparece en la parte derecha de la Figura 3.10.1. Este punto revela información acerca de la componente continua de la imagen original (cuanto mayor sea la

⁶ En realidad implementa un algoritmo *FHT* bidimensional (*Fast Harley Transform*, o Transformada Rápida de Harley)

componente continua de la imagen original, mayor será el brillo de este píxel), y como se puede apreciar, tiene un radio de 512 píxeles por ciclo y un ángulo de 270° . Hay que tener en cuenta que la imagen original, lena.jpg, tiene unas dimensiones de 512x512 píxeles, y de ahí que la componente continua sea de 512 píxeles por ciclo (es decir, cambios en el brillo cada ancho o cada alto de la imagen, o lo que es lo mismo, cambios nulos en el brillo, uniformidad). Conforme se aleja el cursor del centro de la imagen de densidad espectral de potencia se encuentran puntos correspondientes a frecuencias espaciales mayores, es decir, corresponden a cambios de brillo cada menos píxeles, y de ahí que el valor del radio en píxeles por ciclo se reduzca. En cuanto al ángulo de 270° , por tratarse del punto central, el ángulo puede tomar cualquier valor. Para los demás puntos de la imagen de densidad espectral de potencia, el ángulo se mide en sentido *CCW* (*Counter Clockwise*, o contrario al avance de las agujas del reloj) y desde el eje de abscisas positivo.

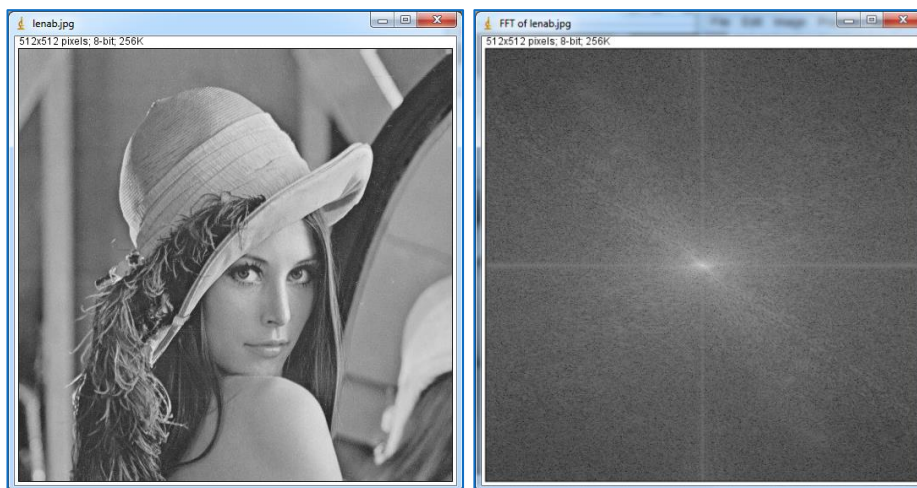


Figura 3.10.1. Izquierda: lena.jpg. Derecha: imagen de tipo 8-bit con la densidad espectral de potencia de la imagen lena.jpg, obtenida con la herramienta *FFT*

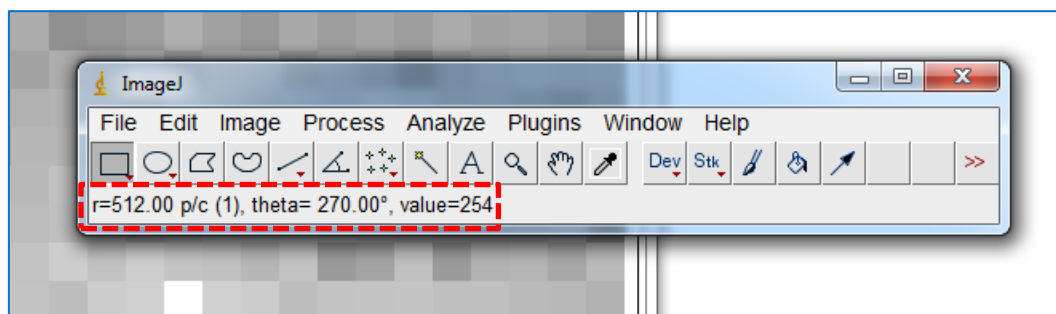


Figura 3.10.2. Barra de estado de ImageJ cuando se desplaza el cursor sobre una imagen con información sobre el espectro de otra imagen. Destacadas en rojo, las coordenadas polares del píxel al que apunta el cursor (el píxel central en ese caso)

También cabe comentar que es posible modificar el espectro de una imagen actuando directamente sobre su imagen de densidad espectral de potencia (aunque estas operaciones, al igual que otras operaciones de

este submenú, en realidad afectan a la Transformada de Fourier de tipo *32-bit* adjunta, que permanece oculta). En primer lugar, es posible filtrar aquella parte del espectro que se quiere hacer desaparecer por completo. Para ello se construye una máscara seleccionando con una herramienta de selección la parte del espectro a eliminar; seguidamente, se elige como color principal el negro absoluto con la herramienta *Color Picker* (submenú *Color*, menú *Image*); y para terminar, se rellena de negro la selección con el comando *Fill* (rellenar) en el menú *Edit*. Para obtener la imagen filtrada basta con aplicar la herramienta *Inverse FFT*, que se revisa a continuación, a la transformada de Fourier modificada con la máscara. También es posible dejar pasar aquella parte del espectro que sea de interés y eliminar el resto. La máscara se construye mediante el procedimiento anterior, pero rellinando la selección con blanco absoluto. De esta forma es posible aplicar a la imagen filtros paso alto o paso bajo, como se muestra en la Figura 3.10.3.

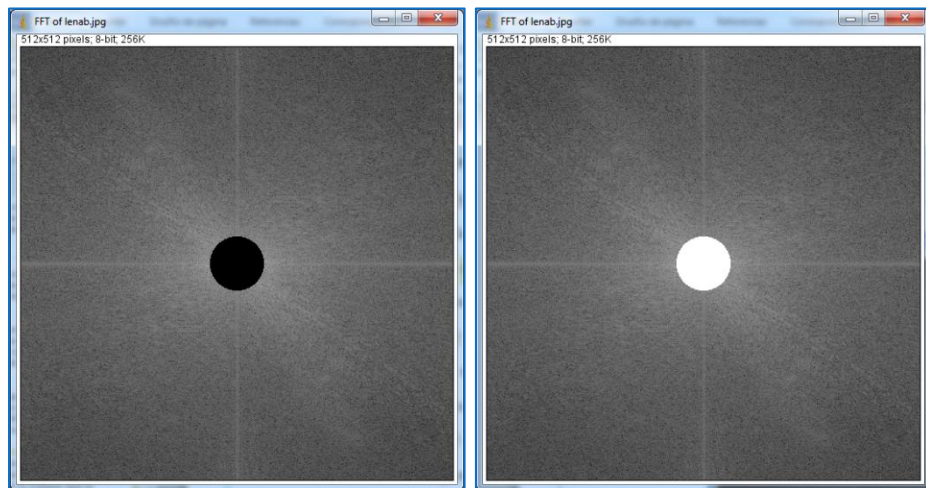


Figura 3.10.3. Izquierda: imagen de densidad espectral de potencia modificada con una máscara para filtrar paso alto la imagen original correspondiente. Derecha: imagen de densidad espectral de potencia modificada con una máscara para filtrar paso bajo la imagen original correspondiente

En las imágenes de densidad espectral de potencia las frecuencias espaciales aparecen representadas por duplicado en puntos opuestos por el centro, por lo que, en principio, cualquier máscara que no estuviera centrada en la componente continua y fuera simétrica estaría incompleta. Sin embargo, ImageJ tiene en cuenta este problema y añade automáticamente los elementos de la máscara necesarios, tal y como se muestra en la Figura 3.10.4:

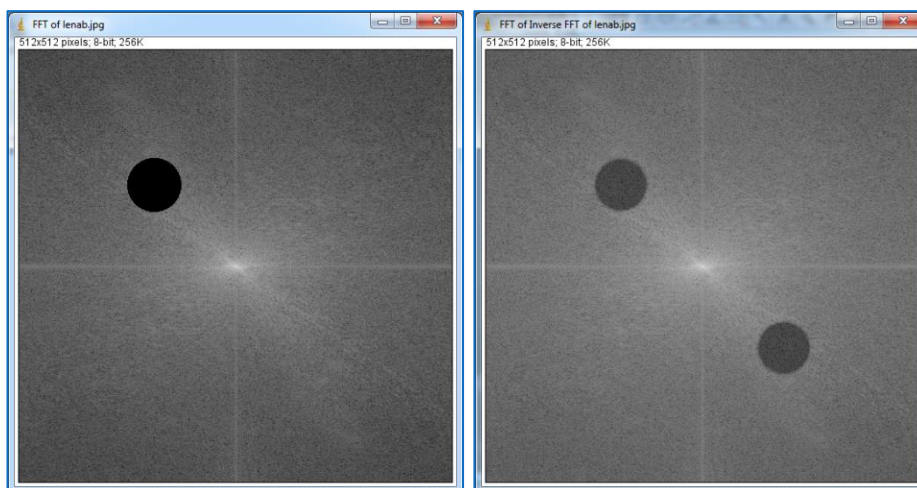


Figura 3.10.4. Imagen de densidad espectral de potencia con una máscara creada por el usuario. Derecha: imagen anterior con la máscara generada automáticamente por ImageJ a partir de la máscara indicada por el usuario

- *Inverse FFT.* Como ya se ha avanzado, esta herramienta permite llevar a cabo la Transformada Inversa de Fourier para obtener imágenes a partir de sus espectros. Esto puede ser deseable a la hora de filtrar imágenes de forma manual, modificando directamente su espectro, como en la Figura 3.10.3 y en la Figura 3.10.4. A continuación, en la Figura 3.10.5 y en la Figura 3.10.6, se muestran unos ejemplos completos de este tipo de operación:

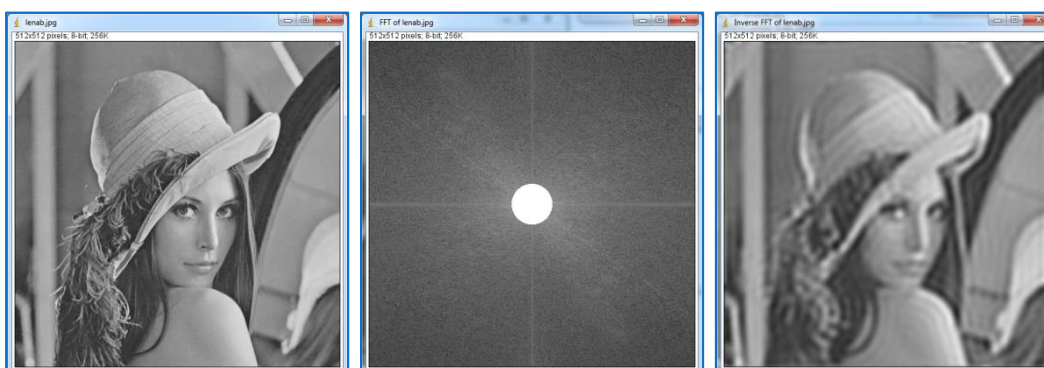


Figura 3.10.5. Izquierda: lena.jpg. Centro: espectro de la imagen anterior con una máscara que implementa un filtro paso bajo. Derecha: imagen resultante de aplicar la Transformada Inversa de Fourier al espectro anterior

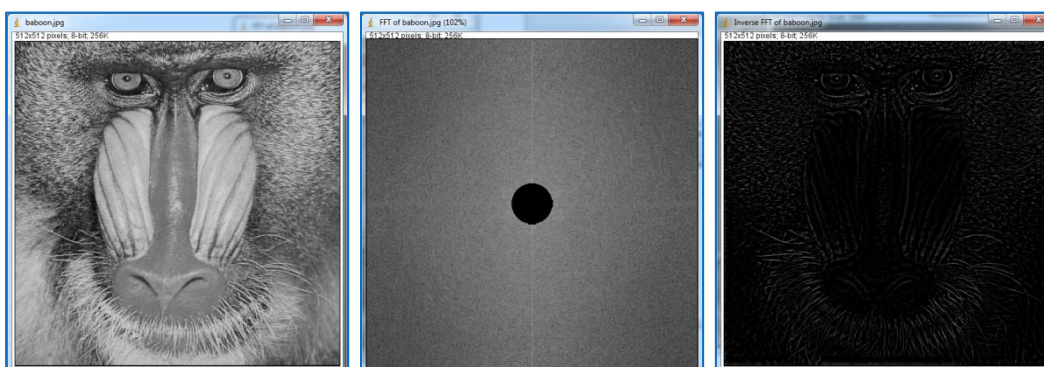


Figura 3.10.6. Izquierda: baboon.jpg. Centro: espectro de la imagen anterior con una máscara que implementa un filtro paso alto. Derecha: imagen resultante de aplicar la Transformada Inversa de Fourier al espectro anterior

- *Redisplay Power Spectrum*. Tras haber operado con las herramientas del submenú *FFT* sobre un espectro, es posible utilizar *Redisplay Power Spectrum* para deshacer todas las operaciones, con lo que se restauran la transformada de Fourier original y se reimprime en pantalla su imagen de tipo *8-bit* asociada con el contenido especificado en el menú emergente *FFT Options*.
- *FFT Options*. En este menú emergente es posible especificar el contenido de la imagen que se genera junto a la transformada de Fourier y que se imprime en pantalla al utilizar la herramienta *FFT*. En el apartado *Display*, se puede elegir entre las siguientes salidas:
 - *FFT window*. Es la salida que se genera por defecto y que ya se ha comentado: imagen de tipo *8-bit* de la densidad espectral de potencia, centrada y procesada con una operación de tipo logaritmo
 - *Raw power spectrum*. Imagen de tipo *32-bit* de la densidad espectral de potencia sin procesar.
 - *Fast Hartley Transform*. Imagen de tipo *32-bit* con la transformada de Fourier⁵.
 - *Complex Fourier Transform*. Pila de imágenes de tipo *32-bits* uno con dos capas, una para la parte real y otra para la parte imaginaria de la transformada de Fourier.

Marcando la casilla *Do Forward Transform* se obtiene de nuevo la transformada de Fourier de la imagen activa con la salida elegida nada más cerrar el menú.

- *Bandpass Filter*. Esta herramienta permite aplicar a la imagen activa un filtro paso banda configurable por el usuario. Concretamente, el usuario puede configurar los siguientes parámetros:
 - *Filter large structures down to*. Literalmente, “Filtrar estructuras grandes como poco hasta” los píxeles indicados por el usuario. Es decir, permite determinar hasta qué punto se eliminan las bajas frecuencias de la imagen (sus partes uniformes, sus estructuras grandes), o lo que es lo mismo, permite especificar el alcance de la componente paso alto del filtro paso banda. Indicando 0 en este campo, la parte paso alto del filtro eliminará el espectro de la imagen desde la componente continua hasta los cambios de brillo cada 0 píxeles (frecuencias infinitamente altas), es decir, eliminará todo el espectro; indicando un número natural inmediatamente superior a las dimensiones de la imagen, la parte paso alto del filtro eliminará frecuencias correspondientes a cambios en el nivel de brillo de tan baja frecuencia espacial que ni siquiera podrán estar presentes en la imagen (se recuerda que para ImageJ la componente continua de la imagen es *d píxeles/ciclo*, siendo *d* sus dimensiones), así que será en realidad paso a todo. Esto se muestra en la Figura 3.10.7:

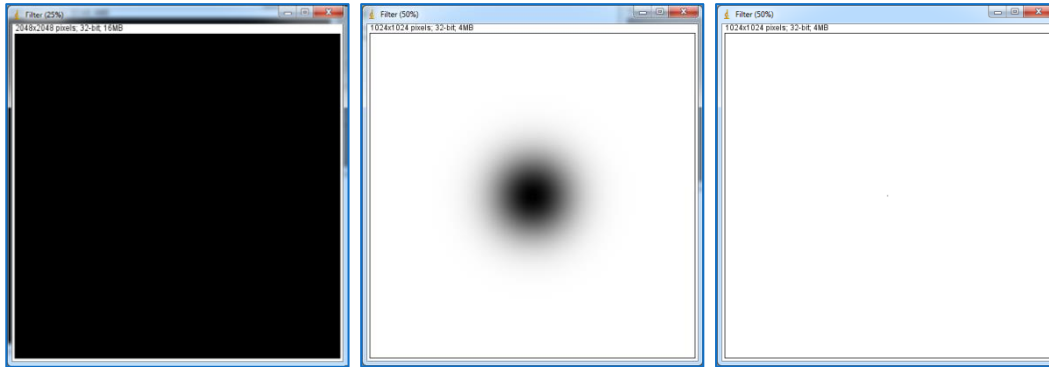


Figura 3.10.7. Izquierda: componente paso alto del filtro paso banda cuando se indica *Filter large structures down to 0 pixels* (no pasa nada). Centro: componente paso alto del filtro paso banda cuando se indica *Filter large structures down to 4 pixels* (pasa la parte alta del espectro). Derecha: componente paso alto del filtro paso banda cuando se indica *Filter large structures down to 513 pixels*, siendo las dimensiones de la imagen activa 512x512 (pasa todo).

- *Filter small structures up to .* . Literalmente, “Filtrar estructuras pequeñas como mucho hasta” los píxeles indicados por el usuario. Es decir, permite determinar hasta qué punto se eliminan las altas frecuencias de la imagen (sus detalles más finos), o lo que es lo mismo, permite especificar el alcance de la componente paso bajo del filtro paso banda. Indicando 0 en este campo, la parte paso bajo del filtro eliminará el espectro de la imagen desde las frecuencias infinitamente altas (cambios en el nivel de brillo cada 0 píxeles) hasta las frecuencias correspondientes a los cambios el nivel de brillo cada 0 píxeles, que son las mismas; es decir, será en realidad paso a todo. Indicando las dimensiones de la imagen en este campo, la parte paso bajo del filtro eliminará el espectro de la imagen desde las frecuencias infinitamente altas (cambios en el nivel de brillo cada 0 píxeles) hasta la componente continua (se recuerda una vez más que para ImageJ la componente continua de la imagen es d píxeles/ciclo, siendo d sus dimensiones), así que eliminará todo. Esto se muestra en la Figura 3.10.8:

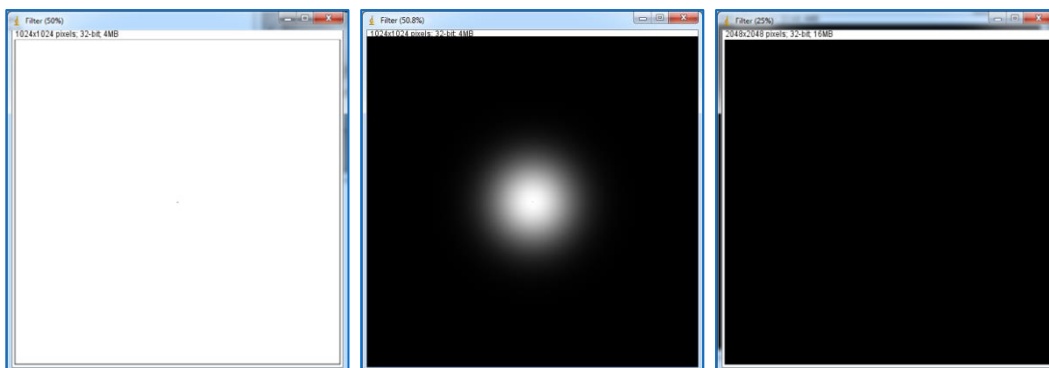


Figura 3.10.8. Izquierda: componente paso bajo del filtro paso banda cuando se indica *Filter small structures up to 0 pixels* (pasa todo). Centro: componente paso bajo del filtro paso banda cuando se indica *Filter small structures up to 4 pixels* (pasa la parte baja del espectro). Derecha: componente paso bajo del filtro paso banda

cuando se indica *Filter small structures up to 512 pixels*, siendo las dimensiones de la imagen activa 512x512 (no pasa nada).

A continuación, en la Figura 3.10.9, se muestra un ejemplo de un filtro paso banda completo generado con *Bandpass Filter*.

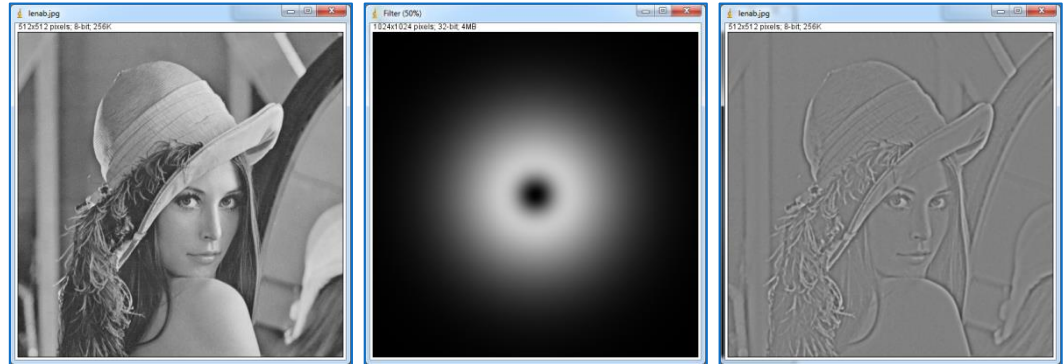


Figura 3.10.9. Izquierda: lena.jpg. Centro: filtro paso banda generado con *Bandpass Filter*. Derecha: lena.jpg filtrada con el filtro paso banda anterior

- *Suppress Stripes*. Esta lista desplegable permite elegir entre *Horizontal*, *Vertical* y *None* para eliminar, respectivamente, las líneas horizontales que en ocasiones generan ciertos escáneres, las verticales o ningún tipo de línea.
- *Tolerance of Direction*. Este campo permite añadir cierta tolerancia al reconocimiento de la dirección de las líneas horizontales o verticales eliminadas según el campo *Suppress Stripes* para, de este modo, eliminar también líneas que tengan una ligera inclinación y no sean completamente horizontales o verticales.
- *Autoscale After Filtering*. Marcando esta casilla, después de llevar a cabo el filtrado paso banda sobre la imagen activa, se escala esta última para que aproveche su rango dinámico (en el caso de las imágenes de tipo 8-bit, por ejemplo, se escala para que su mínimo coincida con 0 y su máximo coincida con 255).
- *Saturate Image when Autoscaling*. En caso de que se haya activado la casilla *Autoscale After Filtering*, activando también *Saturate Image when Autoscaling* se permite que un cierto porcentaje de píxeles se saturen en la operación de autoescalado para aumentar el contraste de la imagen.
- *Display Filter*. Activando esta casilla, además de llevar a cabo el filtrado paso banda sobre la imagen activa se genera una nueva ventana que contiene una representación de la máscara del filtro aplicado.
- *Custom Filter*. Esta herramienta permite utilizar una imagen binaria en el dominio espacial previamente cargada como máscara para filtrar el espectro de la imagen activa. Hay que tener en cuenta que, en este caso, ImageJ no corrige automáticamente la máscara para que sea simétrica con respecto del centro y abarque las frecuencias que aparecen por duplicado.

- *FD Math*. Esta herramienta permite correlar, convolucionar y deconvolucionar en el dominio espacial imágenes del mismo tamaño. Para implementar las operaciones, se transforman las imágenes al dominio de la frecuencia espacial y se multiplican o dividen (y se vuelve a trasladar el resultado al dominio espacial), y de ahí el nombre de *FD Math* (*Frequency Domain Math*, o Matemáticas en el Dominio de la Frecuencia).
- *Swap Quadrants*. Esta herramienta descentra representaciones de transformadas de Fourier de imágenes, de forma que las bajas frecuencias aparezcan en las cuatro esquinas en lugar de en el centro.

3.11. Submenú Filters

En este submenú se muestra una selección de filtros aplicables a la imagen activa:

- *Convolve*. Esta potente herramienta permite filtrar la imagen activa a base de convolucionarla con un *kernel* que el usuario puede especificar manualmente, factor a factor, en la interfaz que se presenta (ver la Figura 3.11.1). Las únicas limitaciones que hay a la hora de definir el *kernel* son que debe ser cuadrado y que sus dimensiones deben ser impares (para que haya una celda central, el ancla, que se hace coincidir con cada píxel). La interfaz además presenta los botones *Open* y *Save* para, respectivamente, cargar y guardar *kernels* en ficheros de texto. También presenta la casilla *Normalize Kernel*, para normalizar el *kernel* dividiendo cada factor por la suma de todos ellos, y la casilla *Preview*, para obtener una vista previa del filtrado de la imagen activa en base a la convolución con el *kernel* especificado.

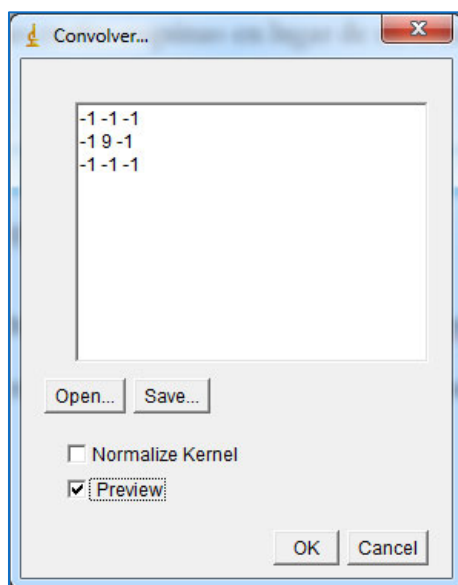


Figura 3.11.1. Interfaz *Convolver* con un *kernel* agudizador

- *Gaussian Blur*. Filtro gaussiano. A través de una interfaz, el usuario puede especificar el valor del radio del filtro (*Radius*) y obtener una vista previa del resultado (*Preview*).
- *Median*. Filtro de mediana.
- *Mean*. Filtro de media.

- *Minimum*. Filtro de mínimo.
- *Maximum*. Filtro de máximo.
- *Unsharp Mask*. Filtro agudizador de tipo *Unsharp Mask*. Estos filtros agudizan la imagen fundamentalmente restando a la imagen original una versión suavizada de ella obtenida con un filtro gaussiano. A través de una interfaz, el usuario puede especificar el valor del radio del filtro gaussiano empleado para obtener la imagen auxiliar, el peso que tiene la imagen auxiliar en la operación (*Mask Weight*) y obtener una vista previa del resultado.
- *Variance*. Filtro de varianza.

A continuación, en la Figura 3.11.2, se muestran varios ejemplos de los efectos de los filtros anteriores:



Figura 3.11.2. De izquierda a derecha y de arriba a abajo: imagen original, imagen filtrada con *Gaussian Blur*, imagen filtrada con *Median*, imagen filtrada con *Mean*, imagen filtrada con *Minimum*, imagen filtrada con *Maximum*, imagen filtrada con *Unsharp Mask* e imagen filtrada con *Variance*

La mayoría de estos filtros tienen su versión tridimensional aplicable a pilas e hiperpilas: *Gaussian Blur 3D*, *Mean 3D*, *Minimum 3D*,...

- *Show Circular Masks*. Genera una pila con las máscaras de diferentes tamaños que emplean los filtros anteriores.

3.12. Submenú Batch

En este submenú se agrupan los comandos relacionados con el procesamiento por lotes, es decir, con la ejecución automática, sin la intervención del usuario, de ciertas tareas repetitivas aplicadas a una gran cantidad de imágenes.

- *Measure*. Se aplica la herramienta *Measure* del menú *Analyze* para analizar todas las imágenes contenidas en la carpeta indicada. Los resultados se muestran en la tabla de resultados.
- *Convert*. Permite aplicar una conversión de formato y un escalado a todas las imágenes contenidas en la carpeta indicada por el usuario.
- *Macro*. Permite aplicar un *macro* a todas las imágenes contenidas en la carpeta indicada. También posibilita especificar el formato de salida y si se lleva a cabo o no un escalado, ofrece fragmentos de código *macro* prefijados para facilitar la elaboración del *macro*, y además incorpora botones para guardar, cargar y testear el *macro* en una imagen individual antes de empezar la operación global.
- *Virtual Stack*. Permite manejar pilas virtuales, que son pilas de imágenes que no residen en la memoria RAM, sino en disco. Su funcionamiento y su interfaz son los mismos que los de la herramienta anterior *Macro*.

3.13. Image Calculator

Esta herramienta permite llevar a cabo las siguientes operaciones aritméticas y lógicas entre los píxeles de las dos imágenes que se indiquen: suma, resta, multiplicación, división, *AND*, *OR*, *XOR*, mínimo, máximo, media aritmética y diferencia (valor absoluto de la resta), todas píxel a píxel (*Add*, *Subtract*, *Multiply*, *Divide*, *AND*, *OR*, *XOR*, *Min*, *Max*, *Average* y *Difference* respectivamente). También aparecen las operaciones *Copy*, que directamente copia los píxeles de la segunda imagen, y *Transparent-zero*, que combina las imágenes considerando que los píxeles de la segunda imagen que tienen un nivel de brillo 0 son totalmente transparentes (y por tanto dejan pasar los píxeles de la primera imagen que ocupan las mismas posiciones), mientras que el resto son opacos (y aparecen sin ninguna modificación).

3.14. Subtract Background

Esta herramienta permite corregir imágenes con fondos que son bastante uniformes pero que presentan cambios suaves de nivel (como puede ocurrir en imágenes con iluminación irregular) para que tengan fondos completamente uniformes y conserven el resto de sus características. Se implementa con un algoritmo de tipo *Rolling Ball*, o bola rodante. Estos algoritmos se denominan así porque su fundamento es el siguiente: en primer lugar, se concibe la imagen como una superficie en la que la tercera dimensión de los puntos viene determinada por el nivel de brillo de los píxeles correspondientes, de forma que los píxeles más brillantes se asocian a puntos de la superficie más altos y viceversa; en este escenario, cada línea de la imagen con fondo cambiante correspondería a un perfil como el que se muestra en la Figura 3.14.1. Por debajo de la curva de este perfil, pero siempre en contacto con éste, se hace rodar un circunferencia imaginaria de un radio lo suficientemente grande para que no se introduzca por la base de los picos del perfil correspondientes a los objetos.

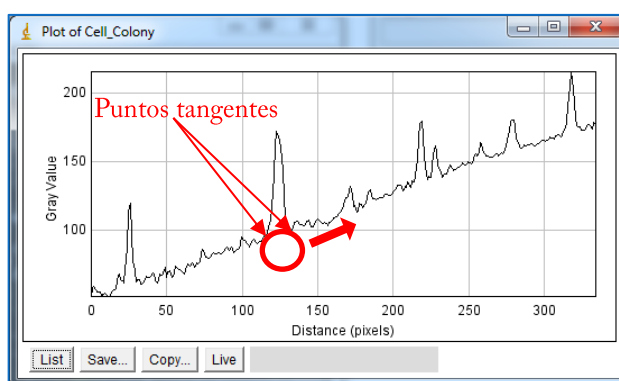


Figura 3.14.1. Perfil de una línea de una imagen con objetos sobre un fondo cambiante

Conforme avanza la circunferencia, entra en contacto con ciertos puntos del perfil, cuya posición se registra. Estos son los puntos que, en la imagen procesada, pasarán a formar parte del fondo, y a los que por tanto se les asigna el mismo nivel de brillo para que el fondo de la imagen resultante sea completamente uniforme. En la Figura 3.14.2, se muestra el mismo perfil de la imagen anterior tras ser procesada con *Subtract Background*:

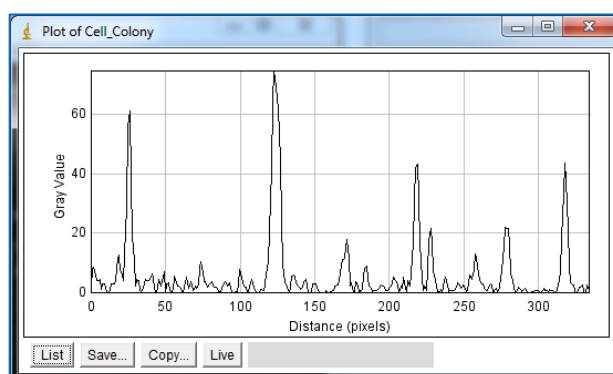


Figura 3.14.2. Perfil de una línea de una imagen con objetos sobre un fondo cambiante tras haberla procesado con *Subtract Background*

A continuación, en la Figura 3.14.3, se muestra el efecto de la herramienta *Subtract Background* sobre una imagen con objetos claros y un fondo oscuro con cambios graduales en el nivel de brillo:

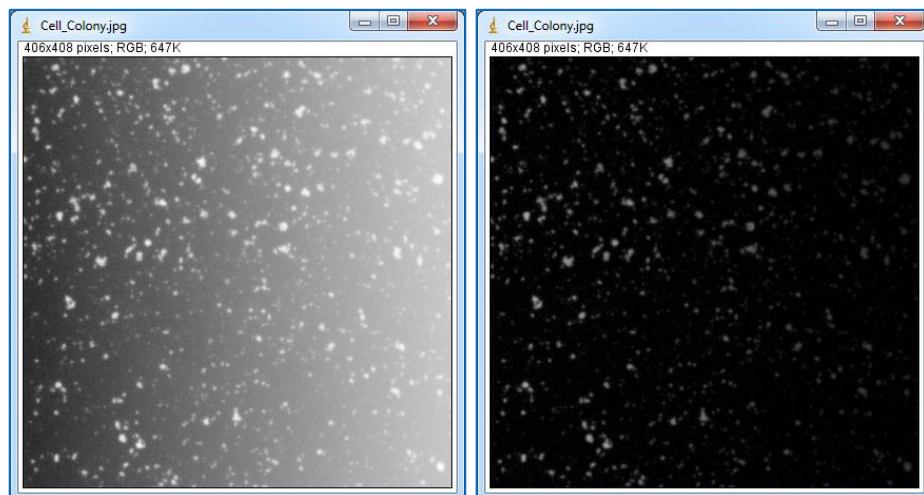


Figura 3.14.3. Izquierda: imagen con objetos claros y un fondo oscuro con cambios graduales en el nivel de brillo. Derecha: imagen anterior tras ser corregida con *Subtract Background*

Para configurar los parámetros fundamentales de la operación, se presenta al usuario un menú con los siguientes campos:

- *Rolling ball radius*. Permite determinar el radio de la esfera empleada en la operación. Como ya se ha avanzado, tiene que ser lo suficientemente grande para que la esfera no entre por la base de los picos correspondientes a los objetos brillantes de la imagen. Por otro lado, una esfera demasiado grande también puede causar problemas en la operación y originar un resultado insatisfactorio.
- *Light background*. Como se puede apreciar en los párrafos anteriores, por defecto se consideran fondos oscuros y objetos claros. Para aplicar la herramienta a imágenes con fondos claros y objetos oscuros, se ha de marcar esta casilla. De esta forma, el algoritmo lo tiene en cuenta y para detectar los puntos del nuevo fondo, en vez de deslizar la esfera por debajo de una superficie con picos, la deja rodar por encima de una superficie con valles. Además, genera un fondo claro.
- *Separate colors*. Aplicable a imágenes en color de tipo *RGB Color*. Marcando esta casilla, la operación se aplica exclusivamente al brillo (no se tienen en cuenta ni se modifican el tono y la saturación).
- *Create background*. Marcando esta casilla, la salida de la operación no es la imagen con el fondo uniforme, sino el fondo desigual extraído.
- *Sliding paraboloid*. Marcando esta casilla se deja de utilizar una esfera y se pasa a utilizar un paraboloide. Es una mejora que generalmente resulta más eficiente y genera resultados más fiables.

- *Disable smoothing.* Antes de llevar a cabo el algoritmo de la bola rodante, se filtra la imagen paso bajo para eliminar detalles que pudieran complicar la operación. Al activar esta casilla, se deja de hacer dicho filtrado, y el algoritmo se lleva a cabo sobre la imagen inalterada.
- *Preview.* Proporciona una vista previa de la operación.

En la Figura 3.14.4 se muestra la interfaz con los campos mencionados.

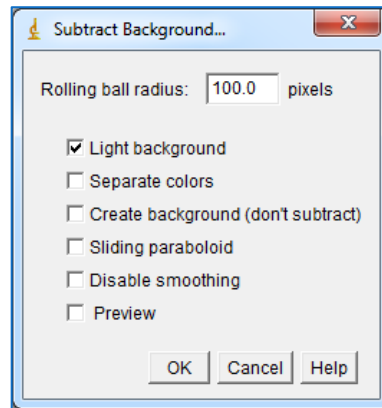


Figura 3.14.4. Menú *Subtract Background*

3.15. Repeat Command

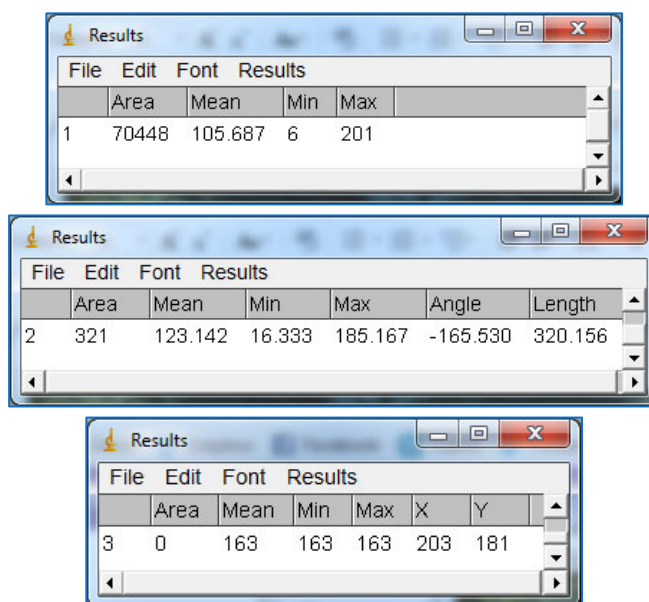
Se ejecuta de nuevo el último comando utilizado.

4. MENÚ ANALYZE

El menú *Analyze* contiene herramientas relacionadas con el análisis de las imágenes.

4.1. Measure

Esta herramienta muestra en la tabla de resultados una serie de medidas realizadas sobre la selección de área, lineal o puntual que haya sobre la imagen activa, o sobre la propia imagen activa en ausencia de selecciones. Vale para cualquier tipo de imagen, aunque en el caso de las imágenes de tipo *RGB Color* se considera para los cálculos el brillo de los píxeles, el cual se obtiene a través de la media aritmética sin ponderar de sus componentes RGB o a través de la expresión (2.1.3) en el caso de que la opción *Weighted RGB Conversions* (apartado *Conversions* del submenú *Options*, menú *Edit*) se encuentre marcada.



The figure shows three screenshots of the 'Results' window in ImageJ, each displaying a different table of measurements:

- Top screenshot (Area selection):** The table has columns: File, Edit, Font, Results, Area, Mean, Min, Max. The data row shows: 1, 70448, 105.687, 6, 201.
- Center screenshot (Line selection):** The table has columns: File, Edit, Font, Results, Area, Mean, Min, Max, Angle, Length. The data row shows: 2, 321, 123.142, 16.333, 185.167, -165.530, 320.156.
- Bottom screenshot (Point selection):** The table has columns: File, Edit, Font, Results, Area, Mean, Min, Max, X, Y. The data row shows: 3, 0, 163, 163, 163, 203, 181.

Figura 4.1.1. Arriba: tabla de resultados con las medidas de una selección de área. Centro: tabla de resultados con las medidas de una selección lineal. Abajo: tabla de resultados con las medidas de una selección puntual

Es posible calcular muchos parámetros estadísticos de la imagen o de las selecciones. Para que éstos se muestren en la tabla de resultados al seleccionar la herramienta *Measure* antes hay que especificarlos (marcando la correspondiente casilla) en el menú emergente *Set Measurements*:

- *Area*. En el caso de las selecciones de área, con este parámetro se indica el área de la selección como el número de píxeles que incluye, sea cual sea la forma de la selección (si la imagen esta calibrada, en vez de píxeles se utilizan las unidades correspondientes). Si se trata de una selección lineal este parámetro vale el producto de la longitud de la línea y su grosor. Por último, en el caso de las selecciones puntuales este parámetros siempre vale 0.

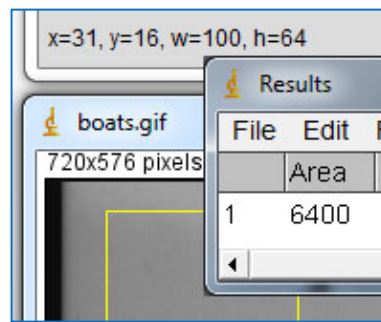


Figura 4.1.2. Selección de ancho $w=100$ y alto $h=64$ junto con la tabla de resultados, en la que se indica que se ha seleccionado un área de $64 \times 100 = 6400$ píxeles cuadrados

- *Mean gray value.* Nivel de brillo medio, calculado a partir de la media aritmética de los niveles de brillo de todos los píxeles considerados:

$$\frac{1}{N} \sum_{i=1}^N u_i \quad (4.1.1)$$

donde N es el número de píxeles considerados y u_i el nivel de brillo del píxel i . Evidentemente, este parámetro da una idea del brillo de las regiones seleccionadas (ver Figura 4.1.3):

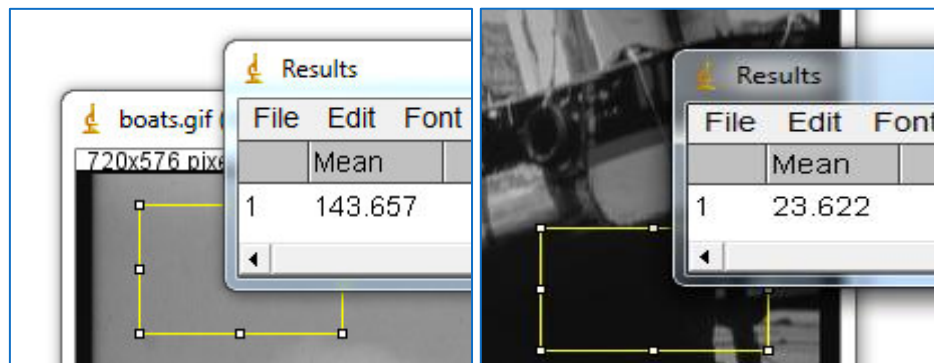


Figura 4.1.3. Izquierda: selección de una región clara junto a su nivel de brillo medio: 144. Derecha: selección de una región oscura junto a su nivel de brillo medio: 24

Si la imagen ha sido calibrada previamente el resultado se da en las unidades escogidas.

- *Standard deviation.* Desviación estándar de los niveles de brillo de los píxeles considerados, calculada a partir de la siguiente expresión:

$$\sqrt{\frac{\sum_{i=1}^N (\bar{u} - u_i)^2}{N - 1}} \quad (4.1.2)$$

donde \bar{u} es el nivel de brillo medio de los píxeles considerados. Como este parámetro estadístico está relacionado con la dispersión, en el caso de las imágenes da una idea del contraste de la imagen. Una imagen uniforme, sin contraste, tendría una desviación estándar de 0, al contrario de lo que sucede con una imagen con píxeles con niveles de brillo muy dispares. Esto se muestra en la Figura 4.1.4.

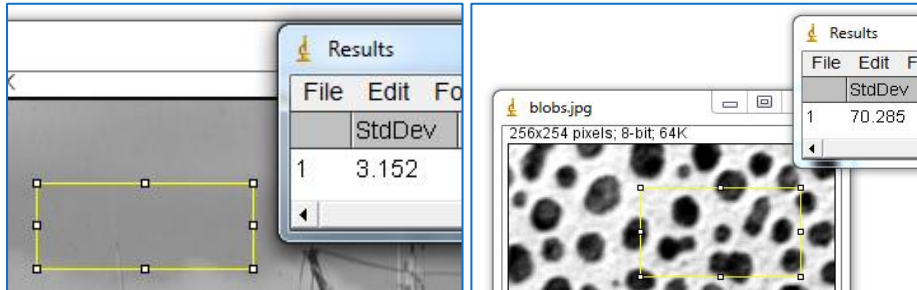


Figura 4.1.4. Izquierda: selección de una región muy uniforme junto a su desviación estándar: 3. Derecha: selección de una región con más contraste que la anterior junto a su desviación estándar: 70

- *Modal grey value.* Nivel de brillo más repetido entre los píxeles considerados.
- *Min & max gray value.* Niveles de brillo máximo y mínimo de entre los niveles de brillo de los píxeles considerados.

Centroid. Es el centroide, o centro de simetría de la selección en cuanto a figura geométrica. En la Figura 4.1.5 se muestra una selección triangular y su centroide señalado sobre ella. Por tratarse de un triángulo, su centroide es el la intersección de sus medianas.

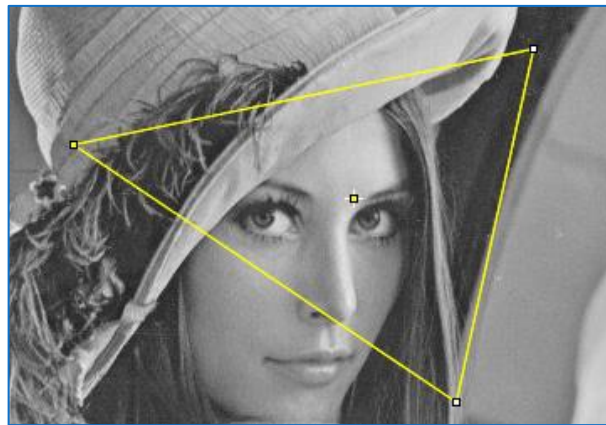


Figura 4.1.5. Selección triangular sobre una imagen y su centroide marcado en medio

- *Center of mass.* Centro de masas que tendría la selección si se considerara el conjunto de sus píxeles como un sistema de partículas que en vez de masa tienen nivel de brillo. Se puede decir que el sistema formado por todo el brillo concentrado en un píxel en el centro de masas es, desde este punto de vista, un sistema equivalente al original. El centro de masas se calcula como sigue:

$$r_{c.m.} = \frac{1}{U} \sum_i u_i r_i \quad (4.1.3)$$

donde $r_{c.m.}$ es el vector de posición del centro de masas, U es el brillo total de los píxeles del sistema, u_i es el brillo de cada píxel del sistema y r_i es el vector de posición de cada píxel del sistema. Sin embargo, ImageJ no ofrece el vector de posición del centro de masas, sino sus coordenadas (ver Figura 4.1.6).

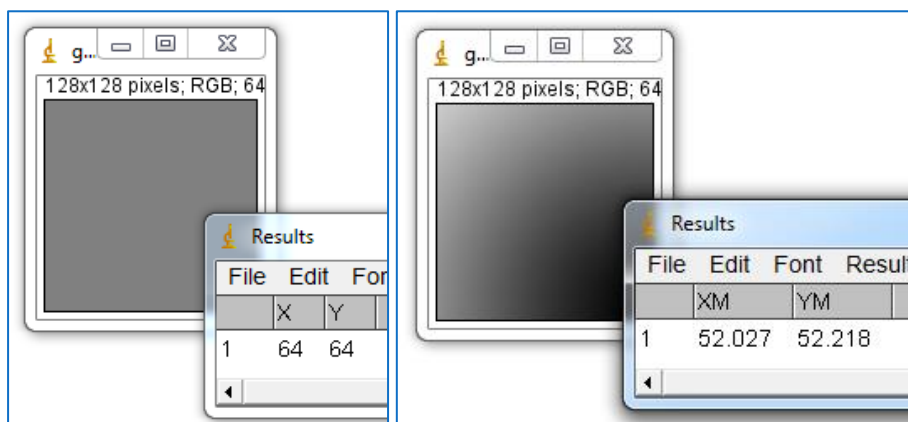


Figura 4.1.6. Izquierda: imagen uniforme de 128x128 y su centro de masas en $x=64$, $y=64$. Derecha: imagen con un degradado de gris y su centro de masas en $x=52,027$, $y=52,218$

En la Figura 4.1.6 se aprecia una imagen uniforme y su centro de masas, que obviamente está en el medio, puesto que los vectores de posición se anulan 2 a 2 por tener el mismo peso. También se observa una imagen con un degradado de gris que va casi desde el blanco en la esquina superior izquierda hasta el negro en la esquina inferior derecha. En este caso el centro de masas se sitúa más cerca de la esquina superior izquierda, esto es, más cerca de la zona más brillante, cuyos píxeles tienen por tanto vectores de posición que se toman en cuenta con un peso mayor en el cálculo del centro de masas.

- *Perimeter*. Perímetro de la selección. En el caso de las selecciones lineales, se indica la longitud de la selección. En cuanto a las selecciones puntuales, su perímetro siempre vale 0.
- *Bounding rectangle*. Rectángulo circunscrito a la selección o, en su defecto, a toda la imagen. En la ventana de resultados se proporcionan las dimensiones de dicho rectángulo, así como las coordenadas de su esquina superior izquierda.
- *Fit ellipse*. Elipse inscrita a la selección o, en su defecto, a toda la imagen. En la ventana de resultados se proporcionan las dimensiones del eje mayor y del eje menor, así como el ángulo que forma el primero con la horizontal.
- *Shape descriptors*. Proporciona varios “descriptores de forma”:
 - *Circularity (Circ)*. “Circularidad”. Este parámetro toma valores entre 1, que es el valor que tomaría un círculo perfecto, y 0, que es el valor que tomaría un “anticírculo” (los valores cercanos a 0 corresponden a formas muy alargadas, casi líneas rectas). Se calcula mediante la siguiente expresión:

$$Circ = 4\pi \frac{area}{perimetro^2} \quad (4.1.4)$$

Evidentemente, en el caso del círculo,

$$Circ = 4\pi \frac{\pi r^2}{(2\pi r)^2} = 4\pi \frac{\pi r^2}{4\pi^2 r^2} = 1 \quad (4.1.5)$$

Por el contrario, las formas muy estiradas tendrán un gran perímetro y un área reducida, lo que implica que su circularidad tenderá a 0.

- *Aspect ratio (AR)*. Relación de aspectos. Es la relación de aspectos de la elipse inscrita a la selección o a la imagen, es decir,

$$AR = \frac{eje\ mayor}{eje\ menor} \quad (4.1.6)$$

Obviamente, es mayor o igual que uno. Éste último caso corresponde a la circunferencia, y el primero, una relación de aspectos muy alta, corresponde de nuevo a figuras muy alargadas. Por eso, como se ve a continuación, a partir de la relación de aspectos se puede definir otro parámetro relacionado con la circularidad: la “redondez”.

- *Roundness (Round)*. “Redondez”. Se define como la inversa de la relación de aspectos, lo cual tiene sentido si se tiene en cuenta la definición anterior. Se calcula como sigue:

$$Round = 4 \frac{area}{\pi \cdot eje\ mayor^2} \quad (4.1.7)$$

Considerando que el área de una elipse es el producto de π y el producto de sus semiejes,

$$\begin{aligned} Round &= 4 \frac{1}{\pi \cdot eje\ mayor^2} \frac{eje\ mayor}{2} \frac{eje\ mayor}{2} \pi = \\ &= \frac{eje\ menor}{eje\ mayor} = \frac{1}{AR} \end{aligned} \quad (4.1.8)$$

Teniendo en cuenta que es la inversa de la relación de aspectos, y que ésta variaba entre ∞ y 1, la redondez varía entre 0 y 1.

- *Solidity*. Solidez. Es la relación entre el área de la selección y el área de la forma convexa que circunscribe a la primera:

$$Solidity = \frac{area}{area\ convexa} \quad (4.1.9)$$

Lógicamente, varía entre 0 y 1, puesto que el área convexa que circunscribe a la primera es mayor o igual que ella. Da una idea de como de ondulado es el perímetro de la selección: cuantas más concavidades haya en la forma de la selección, menor será el área en comparación con el área convexa

y por tanto menor será su solidez, que tenderá a 0. Por el contrario, una selección sin concavidades tendrá una solidez de valor 1.

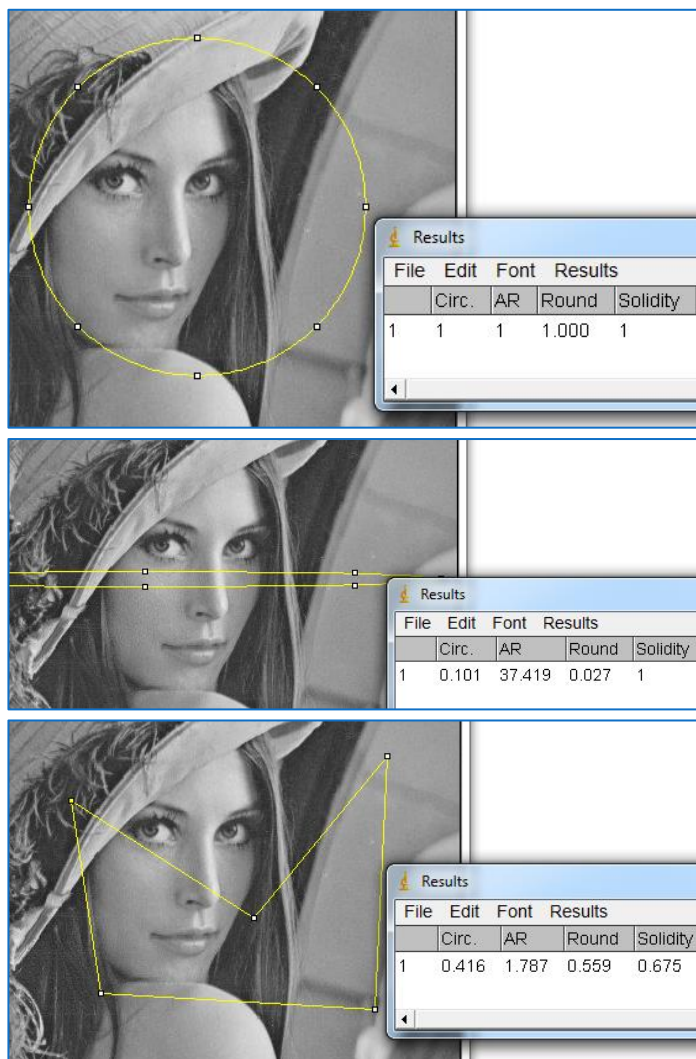


Figura 4.1.7. Ejemplos de diferentes tipos de selecciones junto a sus descriptores de forma

- *Feret's diameter*. Diámetro de Feret. Utilizado en medidas de partículas. Es la distancia que existe entre dos tangentes a la partícula, paralelas entre sí y perpendiculares al eje de medida. En la ventana de resultados se proporcionan el Diámetro de Feret (*Feret*), el ángulo del eje de medida (*Feret Angle*) y las coordenadas del punto donde empieza este último (*FeretX* y *FeretY*). También se proporciona el diámetro mínimo (*MinFeret*). En la Figura 4.1.8 se muestra un ejemplo de la medida del Diámetro de Feret de una partícula con ImageJ:

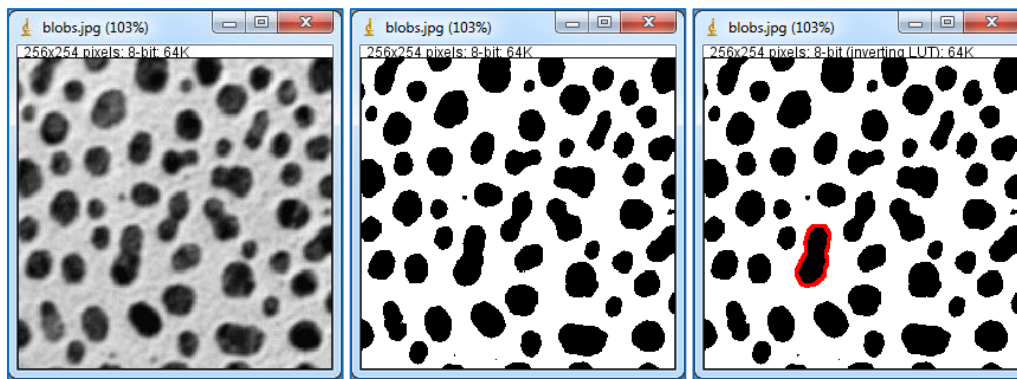


Figura 4.1.8. Primera parte del proceso de medida del Diámetro de Feret en ImageJ

Como muestra la Figura 4.1.8, se parte de la imagen de unas partículas, una de las cuales se va a someter a la medida del Diámetro de Feret. Para poder hacer la selección de forma cómoda, se umbraliza la imagen con las herramientas que se han comentado en apartados anteriores de forma que se conserven las dimensiones de las partículas. Después se utiliza la herramienta *Wand tracing tool* y, haciendo *click* sobre la partícula de interés, inmediatamente se traza una selección perfecta en torno a ella (en el menú emergente *Properties* del submenú *Selection*, menú *Edit*, se le da grosor y se tiñe de rojo para que se puede apreciar más cómodamente). Por último, basta con seleccionar la opción *Feret's Diameter* en *Set Measurements* y hacer *click* en *Measure*.

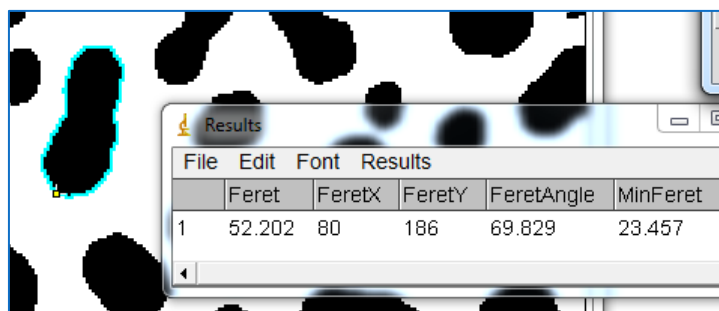


Figura 4.1.9. Diámetro de Feret de una partícula. Se señala la posición $x=80$ e $y=186$, que es el punto donde comienza el eje de medida

Si se hubiera tratado de una imagen debidamente calibrada se habría calculado el Diámetro de Féret real de la partícula en las unidades adecuadas.

- *Integrated density*. Densidad integrada. Es el brillo total de la selección o de la imagen caculado a partir de la siguiente expresión:

$$IntDen = Mean\ Gray\ Value \times Area \quad (4.1.10)$$

Como la media es sensible a las calibraciones de la imagen, la densidad integrada (*IntDen*) también lo es. Por eso se proporciona un parámetro adicional: *Raw integrated density* o Densidad integrada bruta

(*RawIntDen*). Este parámetro es independiente de calibraciones porque se calcula sumando directamente los valores del brillo de todos los píxeles:

$$RawIntDen = \sum_i u_i \quad (4.1.11)$$

Evidentemente, si la imagen no se ha calibrado, estos dos parámetros coinciden.

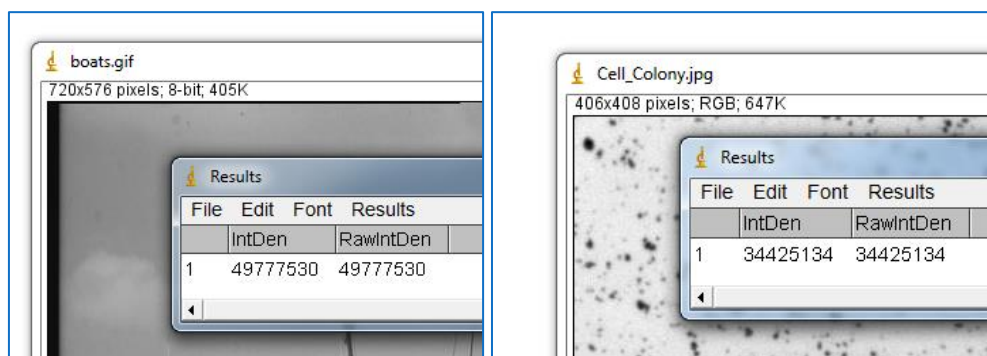


Figura 4.1.10. Dos imágenes de diferente brillo y distinto tamaño junto a sus densidades integradas

En la Figura 4.1.10 se pueden ver dos imágenes junto a sus densidades integradas. La de la parte izquierda es más oscura y más grande, y la de la parte derecha es más clara y más pequeña. La media sería mayor en ésta última por ser más brillante, pero no así la densidad integrada, que es más grande en la imagen de la izquierda por ser más grande aunque sea más oscura (aunque de menor valor, se tienen en cuenta más términos en el sumatorio).

- *Median*. Mediana del conjunto de niveles de brillo de todos los píxeles. Es el nivel de brillo que, al ordenar los valores de todos los píxeles de menor a mayor, ocupa la posición central.
- *Skewness*. Asimetría. Es una medida estadística de la asimetría del conjunto de los niveles de brillo de los píxeles. Se calcula en base al tercer momento estándar en torno a la media.
- *Kurtosis*. Curtosis. Medida estadística de la forma del conjunto de los niveles de brillo de los píxeles. Da una idea de la concentración de valores en torno a la media. Se calcula en base al cuarto momento estándar en torno a la media.
- *Area fraction*. Fracción de área. Cuando se mide sobre imágenes umbralizadas, muestra el porcentaje de píxeles umbralizados:

$$\%Area = 100 \frac{area\ umbralizada}{area\ total} \quad (4.1.12)$$

En el caso de imágenes sin umbralizar, muestra el porcentaje de píxeles distintos de cero.

- *Stack position*. Posición de pila. Aplicado a pilas o hiperpilas, muestra la posición que ocupa la selección o imagen actual en la pila.

4.2. Analyze Particles

Esta herramienta permite contar, medir y analizar objetos en imágenes binarias o umbralizadas, o en selecciones hechas sobre ellas. Se basa en avanzar por la imagen o selección de izquierda a derecha y de arriba abajo hasta que se detecta el borde de un objeto, después se delinea dicho objeto (haciendo uso de la herramienta *Wand Tool*) y se analiza (usando la herramienta *Measure*), y una vez hecho esto se hace invisible el objeto y se vuelve a empezar. Este proceso se repite hasta que se alcanza el final de la imagen.

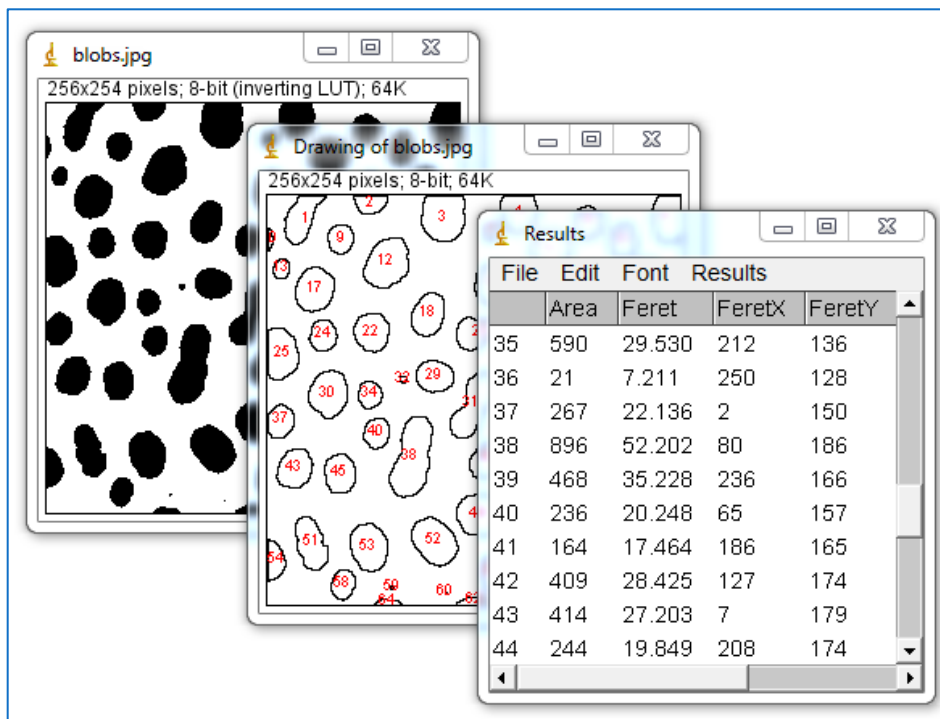


Figura 4.2.1. Análisis de las partículas presentes en la imagen umbralizada blobs.jpg. Junto a ella se muestra el dibujo tipo *Outline* correspondiente y la tabla de resultados con los datos obtenidos tras el análisis

Antes de llevar a cabo el análisis de las partículas se muestra una ventana emergente (ver Figura 4.2.2) que permite configurar diferentes parámetros del proceso:

- **Size.** Este campo de texto permite especificar el rango de tamaños que han de considerarse para el conteo; las partículas con tamaños fuera de este rango se ignoran. Las unidades son “píxeles cuadrados” para imágenes sin calibrar y las unidades que correspondan para las imágenes calibradas. Por defecto toma el valor 0 – *Infinity*, es decir, se consideran todas las partículas, sea cual sea su tamaño. Si se especifica sólo un valor x , entonces se entenderá como $x - \text{Infinity}$.

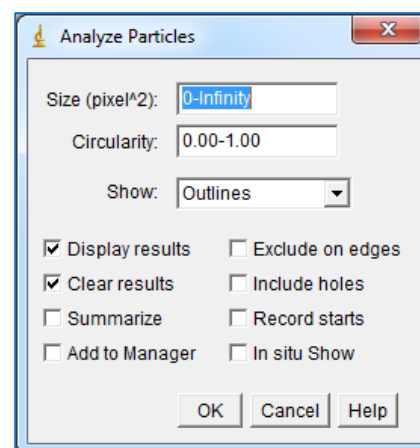


Figura 4.2.2. Menú *Analyze Particles*

- *Circularity*. Este otro campo de texto permite especificar el rango de valores de circularidad (concepto comentado en los párrafos anteriores) que pueden tener las partículas que se van a considerar en el conteo. Se pueden especificar valores desde 0 (partículas muy alargadas) hasta 1 (partículas perfectamente circulares).
- *Show*. Esta lista desplegable permite elegir el tipo de imagen o recubrimiento que se representará una vez terminado el análisis:
 - *Nothing*. No se muestra ninguna imagen.
 - *Outlines*. Se imprime en pantalla una imagen de tipo *8-bit* que contiene los contornos de las partículas numerados (en el orden de análisis).
 - *Bare outlines*. Se representa una imagen de tipo *8-bit* con los contornos de las partículas sin numerar.
 - *Masks*. Imagen de tipo *8-bit* con los contornos de las partículas rellenos con color negro (máscaras) y sin numerar.
 - *Ellipses*. Imagen de tipo *8-bit* que contiene, para cada partícula, la elipse que mejor la sustituye.
 - *Count Masks*. Se genera una imagen de tipo *16-bit* con los contornos de las partículas rellenos. En este caso se rellenan con un gris cuyo nivel de brillo se corresponde con el número de la partícula (aunque éste no se muestra) y se muestran sobre un fondo negro.
 - *Overlay Outlines*. Elimina la información previamente contenida en el recubrimiento de la imagen y la sustituye por los contornos de las partículas. En el recubrimiento también se incluyen etiquetas para numerar las partículas. No se genera ninguna nueva imagen.
 - *Overlay Masks*. Elimina la información previamente contenida en el recubrimiento de la imagen y la sustituye por las máscaras (contornos rellenos con el color elegido para el recubrimiento) de las partículas. En el recubrimiento también se incluyen etiquetas para numerar las partículas. Tampoco se genera ninguna nueva imagen.
- *Display results*. Marcando esta casilla se muestran en la ventana de resultados todas las medidas especificadas en el menú *Set Measurements* para cada partícula.
- *Clear results*. Marcando esta casilla se vacía la lista de resultados en caso de contuviera medidas realizadas anteriormente.
- *Summarize*. Se genera una nueva tabla aparte de la tabla de resultados en la que se muestra la media de todos los valores de todos parámetros elegidos de las partículas de la imagen. Además, se muestra el número de partículas contadas (*Count*), el área que ocupan todas las partículas (*Total Area*), el tamaño medio de las partículas (*Average Size*) y el porcentaje del área total que ocupan las partículas (*%Area*).
- *Add to Manager*. Si se marca esta casilla, tras la medida se añaden las selecciones de las partículas al gestor de regiones de interés (*ROI Manager*).

- *Exclude on edges*. Marcando esta casilla se ignoran en el análisis los objetos que toquen los bordes de la imagen o selección.
- *Include holes*. Al marcar esta casilla, los huecos que presenten las partículas se incluyen por completo en las propias partículas. Por el contrario, si la casilla se encuentra desmarcada, los huecos de las partículas no se toman como partes de la partículas, y por tanto es posible medir partículas dentro de partículas:

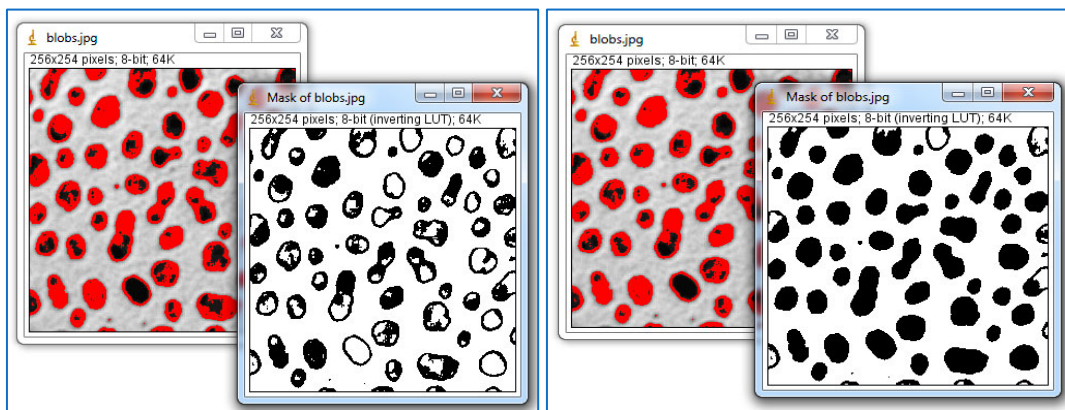


Figura 4.2.3. Izquierda: máscara de las partículas de la imagen sin usar *Include Holes*. Derecha: máscara de las partículas de la imagen usando *Include Holes*

- *In situ Show*. Marcando esta casilla, la imagen original se reemplaza por la imagen generada tras el análisis de partículas: *Outlines*, *Bare Outlines*, *Masks*, *Ellipses* o *Count Masks*.

4.3. Summarize

Esta herramienta calcula la media, la desviación estándar, el máximo y el mínimo del conjunto de los valores de las columnas de la tabla de resultados para todas las columnas y los muestra en la propia tabla:

Results							
File Edit Font Results							
	Label	Area	Feret	FeretX	FeretY	FeretAngle	MinFeret
59		3	2.828	76	244	45	2
60		1	1.414	110	246	135	1
61		80	19.105	170	252	173.991	6
62		89	22.091	117	252	174.806	5
63		52	16.031	227	253	176.424	5
64		49	17.029	66	253	176.634	4
65	Mean	345.572	24.840	123.719	131.094	85.044	16.936
66	SD	216.739	9.522	79.463	75.177	41.768	7.131
67	Min	1	1.414	0	0	6.953	1
68	Max	896	52.202	254	253	176.634	27

Figura 4.3.1. Tabla de resultados después de usar la herramienta *Summarize*

4.4. Distribution

Esta herramienta genera un histograma a partir de todos los valores de la columna de la tabla de resultados que se indique para obtener una representación gráfica de su distribución (frecuencia relativa).

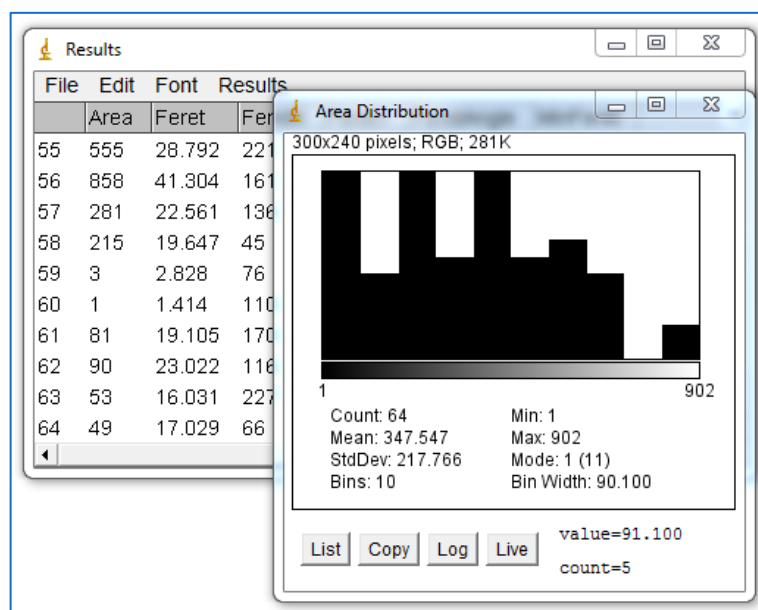


Figura 4.4.1. Tabla de resultados junto a un histograma que representa la distribución de los valores de su primera columna (área)

Bajo el histograma se muestra el número de valores considerados (*Count*), su media (*Mean*), su desviación estándar (*StdDev*), su *máximo* (*Max*) y su *mínimo* (*Min*), y además se muestran otros datos de interés, como el número de columnas utilizadas (*Bins*), el número de unidades que abarca cada columna (*Bin Width*) y la columna que contiene los valores más frecuentes (*Mode*, moda; el valor que sigue a los dos puntos es el extremo izquierdo de la columna en cuestión, y entre paréntesis se muestra el número de veces que aparece alguno de los valores incluidos en dicha columna). En la parte inferior derecha de la ventana se muestran más detalles de la barra sobre la que descansa el cursor. A su izquierda hay cuatro botones: *List*, *Copy*, *Log* y *Live*. *List* despliega una nueva ventana en la que se describe en forma de lista la información del histograma. *Copy* traslada los datos que componen esta lista al portapapeles del sistema. *Log* superpone al histograma original su versión logarítmica.

Antes de calcular el histograma se despliega un menú emergente que permite configurar la operación. A través de la lista desplegable *Parameter* se permite elegir la columna de la tabla de resultados que se quiere utilizar para realizar el histograma. Seleccionando *Automatic binning* se selecciona de forma automática el ancho de barra óptimo. Si no se selecciona, se puede especificar manualmente el número de barras del histograma y el rango de valores del parámetro elegido que se tienen en cuenta en el histograma.

4.5. Label

Utilizando esta herramienta se imprimen sobre la imagen (de forma invasiva) etiquetas con los números de las entradas de la tabla de resultados. Se imprimen en las coordenadas del centroide de las partículas, por lo que es necesario marcar la casilla *Centroid* en *Set Measurements*.

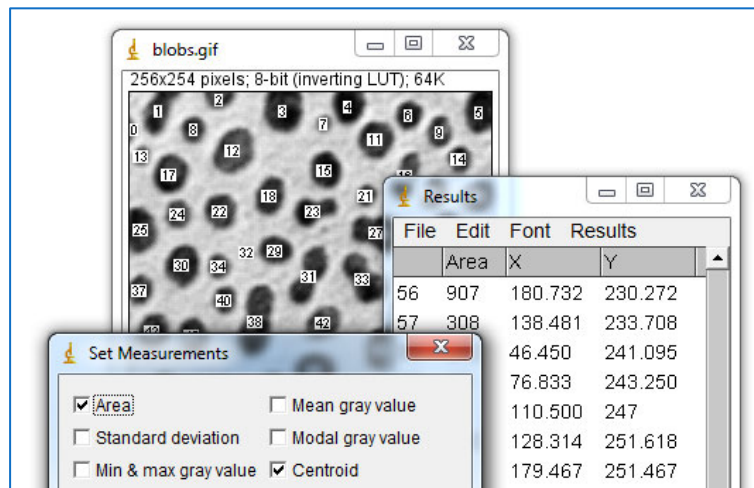


Figura 4.5.1. Imagen con etiquetas junto a la correspondiente tabla de resultados y junto al menú de *Set Measurements*, en el que la opción *Centroid* aparece marcada.

4.6. Clear Results

Con esta opción se vacía la tabla de resultados y se resetea el contador de medidas (las nuevas medidas empiezan a numerarse desde 0 y no desde el número correspondiente a la medida anterior).

4.7. Set Measurements

Como ya se ha explicado, a través de este menú emergente es posible elegir las medidas que se llevan a cabo al utilizar las herramientas *Measure* o *Analyze Particles*: área, nivel medio, desviación estándar, moda,... Además, posibilita la configuración general de la medida a través de las siguientes opciones:

- *Limit to threshold*. Marcando esta opción se limitan las medidas a los píxeles umbralizados de la selección o imagen.
- *Display label*. Al marcar esta opción se añade a la izquierda de la tabla de resultados una columna con el nombre de la imagen sobre la que se ha llevado a cabo cada medida. En el caso de pilas o hiperpilas, se nombra también la capa. Si en vez de una imagen completa se trata de una selección, se anexa la etiqueta de la selección en caso de que la tenga.
- *Invert Y coordinates*. Marcando esta opción se sitúa el origen de coordenadas en la esquina inferior izquierda de la imagen. En caso contrario, se sitúa en la esquina superior izquierda.
- *Scientific notation*. Marcando esta casilla, los resultados de las medidas se muestran en notación científica.

- *Add to overlay*. Seleccionando esta opción, las regiones de interés analizadas se añaden de forma automática al recubrimiento de la imagen.
- *Redirect to*. Esta lista desplegable permite elegir la imagen sobre la que se harán los cálculos al utilizar la herramienta *Measure*. De esta forma es posible hacer una selección sobre una imagen y analizar la región delimitada por dicha selección, pero en otra imagen.
- *Decimal places*. Permite especificar el número de decimales que se usan tanto en la tabla de resultados como en los histogramas.

4.8. Set Scale

Con esta herramienta es posible calibrar espacialmente las imágenes estableciendo una relación entre los píxeles y una unidad de longitud. De este modo es posible hacer medidas espaciales y obtener resultados con valores reales relativos a unidades espaciales. Para calibrar espacialmente una imagen, en primer lugar hay que hacer una selección lineal que se corresponda con una longitud conocida (ver Figura 4.8.1, parte izquierda). Una vez hecho esto, hay que seleccionar la herramienta *Set Scale*, lo cual despliega el menú emergente que se muestra en la parte derecha de la Figura 4.8.1.

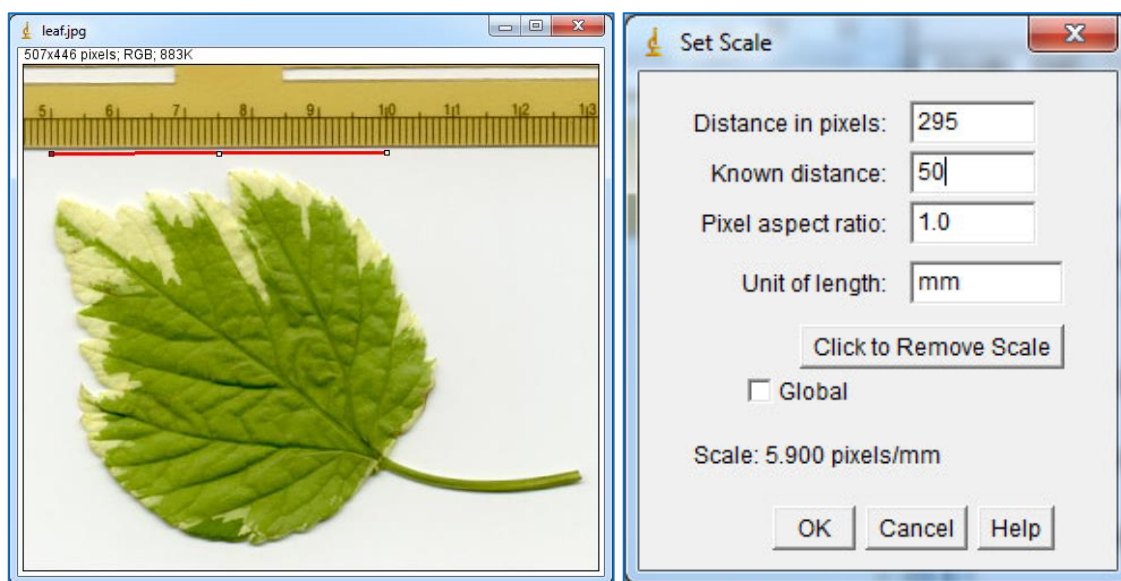


Figura 4.8.1. Izquierda: imagen con una distancia conocida y una selección lineal sobre ella. Derecha: menú emergente *Set Scale*

El siguiente paso consiste en introducir el valor de la distancia conocida y sus unidades en los campos de texto *Known distance* y *Unit of length* respectivamente. Al pulsar el botón *OK* se define la escala de la imagen en base a la selección línea hecha y los valores indicados en los campos *Known distance* y *Unit of length* (el campo *Distance in pixels* se rellena de forma automática).

Una vez se ha calibrado la imagen, todas las medidas espaciales que se efectúen sobre ella arrojarán resultados en la unidad de longitud indicada. Para corroborar que efectivamente se ha calibrado la imagen, basta con mirar la parte superior del marco de la imagen: en las imágenes calibradas, además de indicarse el tamaño en píxeles, se indica el tamaño real:

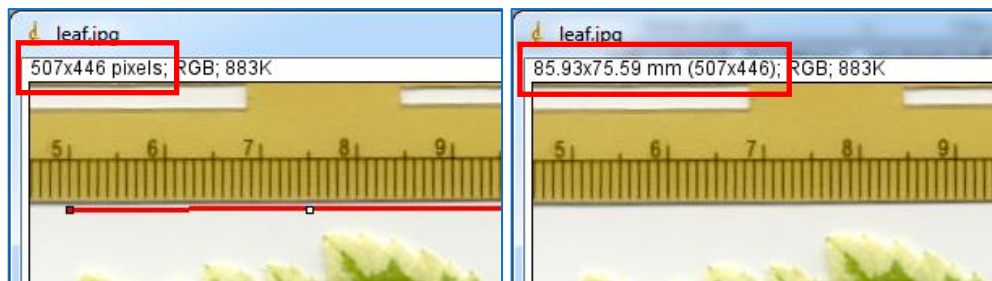


Figura 4.8.2. Izquierda: imagen sin calibrar; destacado en rojo, su tamaño en píxeles. Derecha: imagen calibrada; destacados en rojo, su tamaño real y su tamaño en píxeles

Habiendo calibrado la imagen de la parte izquierda de la Figura 4.8.1, sería posible, por ejemplo, obtener el valor real del área fotosintética (verde) de la hoja. Primero habría que umbralizar adecuadamente la imagen de forma que la región extraída fuera la parte verde de la hoja. En segundo lugar sería necesario configurar la medida en el menú *Set Measurements* marcando al menos la casilla *Area*. A continuación habría que hacer una selección en torno a la hoja para excluir otras regiones umbralizadas que no son de interés. Después habría que elegir la herramienta *Analyze Particles* y configurar el proceso en el menú emergente, eligiendo un rango de tamaños que excluya pequeñas regiones umbralizadas que no son de interés incluidas dentro de la selección, y que al mismo tiempo incluya el área fotosintética (también puede resultar útil definir un intervalo de circularidad válido). También es necesario marcar la casilla *Display results* para obtener el resultado de la medida en la tabla de resultados.

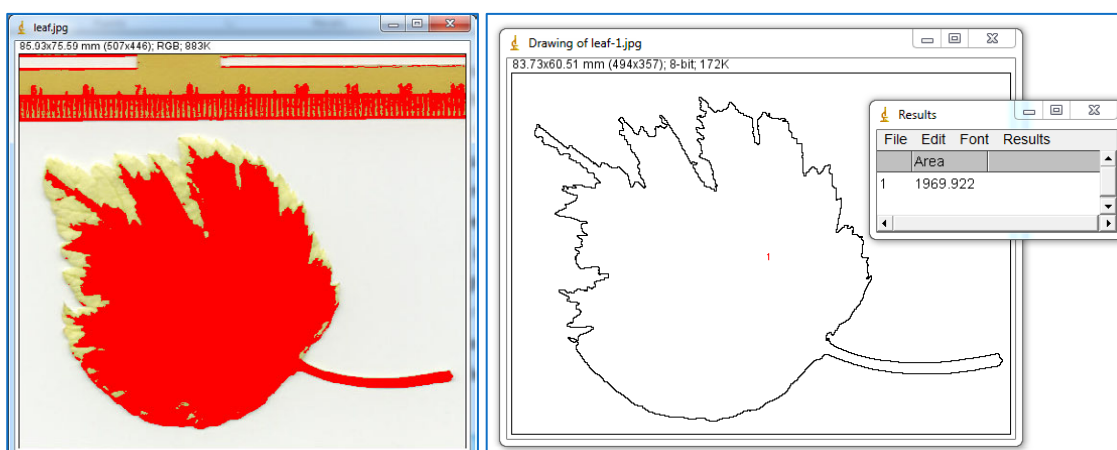


Figura 4.8.3. Izquierda: área fotosintética de una hoja umbralizada. Derecha: contorno del área fotosintética de la hoja junto a la tabla de resultados, en la cual se muestra el valor real del área (1969,922 mm²)

Hay más funcionalidades relativas a la calibración espacial además de las que se han comentado hasta ahora. Por ejemplo, es posible utilizar diferentes calibraciones para las dimensiones vertical y horizontal de una misma imagen. Para ello hay que introducir el valor adecuado (el resultado de dividir ancho entre alto) en el campo *Pixel aspect ratio* en el menú *Set Scale*. También es posible aplicar la calibración a todas las imágenes de la sesión marcando la casilla *Global*. Por último, el botón *Click to Remove Scale* resetea todos los parámetros y elimina la calibración.

4.9. Calibrate

Con esta herramienta es posible calibrar la densidad de las imágenes estableciendo una relación entre el nivel de brillo de los píxeles y una magnitud real: temperatura, elevación, salinidad,... El proceso a seguir para calibrar la densidad de las imágenes es equivalente al proceso de calibración espacial. El primer paso consiste en utilizar las herramientas de selección de área y la herramienta *Measure* configurada adecuadamente para medir los niveles de brillo medio de zonas de la imagen en las que los valores reales de la magnitud a calibrar sean conocidos. Las medidas se almacenan en la tabla de resultados. Este paso se ilustra en la Figura 4.9.1, en la que se mide los valores medios de brillo de las barras de una tabla escalonada calibrada Kodak No. 3 Calibrated Step Tablet, de la cual se conocen los valores de densidad óptica (la magnitud final) de los escalones.

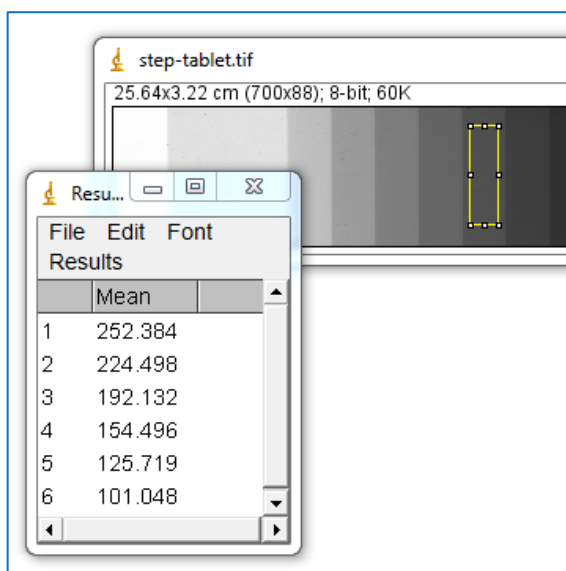


Figura 4.9.1. Primer paso de la calibración de una imagen: medida de los niveles de brillo medios de las regiones de la imagen cuyos valores de la magnitud final son conocidos

Una vez hecho esto se ha de seleccionar la opción *Calibrate*, lo cual despliega el menú de la Figura 4.9.2. La mayor parte de este menú la ocupan dos columnas, y la de la izquierda aparece rellena con los niveles de brillo medio que se han registrado. El siguiente paso consiste en rellenar la columna derecha con los valores reales de la magnitud a calibrar correspondientes a cada uno de los valores medios de brillo que aparecen en la

columna de la izquierda. En este caso son valores de densidad óptica conocidos: 0,00 O.D., 0,06 O.D., 0,20 O.D.,... Para terminar, hay que indicar en el campo *Unit* las unidades de los valores de la columna derecha (en este caso, O.D., *Optical Density*) y elegir en la lista desplegable *Function* una función que se ajuste a la relación que se desea establecer entre el brillo de los píxeles y la magnitud final: relación lineal (*Straight Line*), polinómica de 2º grado (*2nd Degree Polynomial*), polinómica de 3º grado (*3rd Degree Polynomial*),...(en esta ocasión, se elige *Rodbard*) Si se activa la casilla *Show plot*, dicha relación se muestra de forma gráfica en una nueva ventana.

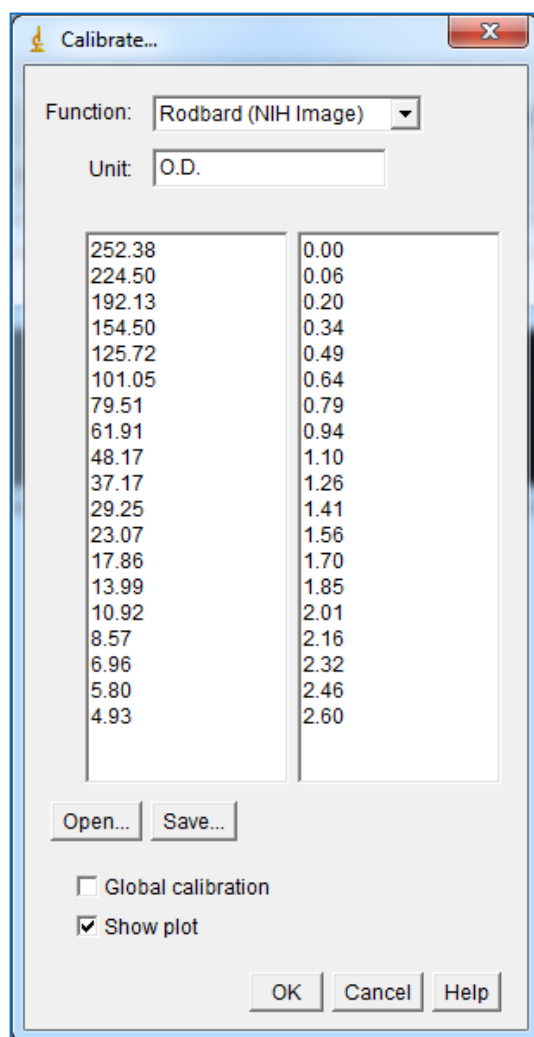


Figura 4.9.2. Menú *Calibrate*

Los botones *Open* y *Save* permiten cargar y guardar la información de las calibraciones en archivos de texto. La casilla *Global calibration* permite extender la calibración a todas las imágenes de la sesión.

4.10. Histogram

Con esta herramienta es posible calcular y representar el histograma de los niveles de brillo de los píxeles contenidos en el área seleccionada o, en su defecto, en la totalidad de la imagen activa: el eje de abscisas

contiene todos los niveles de brillo que puede contener la selección o imagen y el eje de ordenadas contiene las frecuencias absolutas de dichos niveles (las veces que los niveles de brillo de los píxeles de la imagen toman cada valor). Bajo el eje de abscisas se muestra una barra coloreada en la que los niveles de brillo o colores se corresponden con el valor que aparece en el eje justo encima, de tal manera que es posible identificar visualmente la frecuencia relativa de un nivel de brillo o color en la selección o imagen. Un ejemplo de esto se muestra en la Figura 4.10.1:

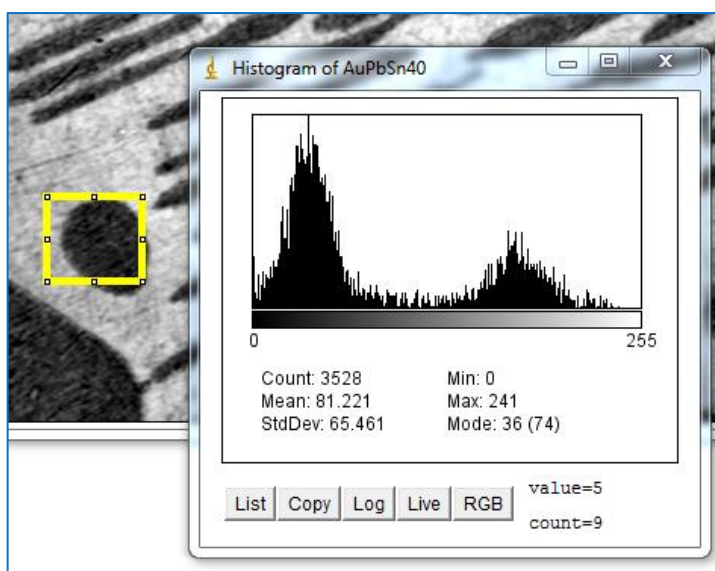


Figura 4.10.1. Selección de una zona oscura de una imagen junto a su correspondiente histograma

En la Figura 4.10.1 aparece un ejemplo de un histograma de niveles de brillo. En este caso corresponde a una selección de área de una zona en la que la mayoría de píxeles son oscuros y hay un pequeño porcentaje de píxeles claros, y de ahí que el grueso del histograma se concentre en la parte izquierda, correspondiente a niveles bajos, mientras que el resto tiene mucho menos contenido.

En la ventana que contiene el histograma se proporciona más información aparte del propio histograma: el número total de píxeles considerados (*Count*), el nivel de brillo medio que tienen estos píxeles (*Mean*), la desviación estándar del conjunto de valores (*StdDev*) y la moda (*Mode*; el valor que se muestra en realidad no es la moda del conjunto de valores, sino el valor que marca el comienzo de la columna que representa a los valores que más aparecen; entre paréntesis se muestra el número de veces que aparece alguno de los valores que representa la columna). También hay una serie de botones que proveen de otras funcionalidades: *List* despliega una nueva ventana en la que se describe en forma de lista la información del histograma; *Copy* traslada los datos que componen esta lista al portapapeles del sistema; *Log* superpone al histograma original su versión logarítmica; *Live* permite mostrar el histograma en directo, en cuyo caso se resalta el botón en rojo (el histograma se recalcula en tiempo real a medida que se mueve la selección de área o se navega entre las capas de una pila). A la derecha de estos botones se muestran dos valores, *Value* y *Count*, que cambian a

medida que se desplaza el cursor sobre el histograma; son, respectivamente, los niveles de brillo y el número de píxeles de la selección o imagen que toman cada valor. Si la imagen o pila activa es una imagen o pila en color se muestra otro botón adicional, *RGB*. En este tipo de imágenes a cada píxel le corresponden tres valores que definen su color de modo que se puede hablar de los canales R, G y B de la imagen. El botón *RGB* permite conmutar entre los histogramas correspondientes a cada uno de los canales, que, por lo tanto, contienen en el eje de abscisas los posibles niveles de rojo, verde o azul, y en el eje de ordenadas, el número de píxeles de la imagen que toman cada uno de dichos niveles. También se muestra un histograma del nivel de brillo, calculando el brillo de cada píxel a partir de la fórmula (2.1.3) en caso de que la opción *Weighted RGB Conversions* del submenú *Options* (menú *Edit*) se encuentre marcada, o a través de la media aritmética de los niveles RGB en caso contrario. Esto se muestra en la Figura 4.10.2.

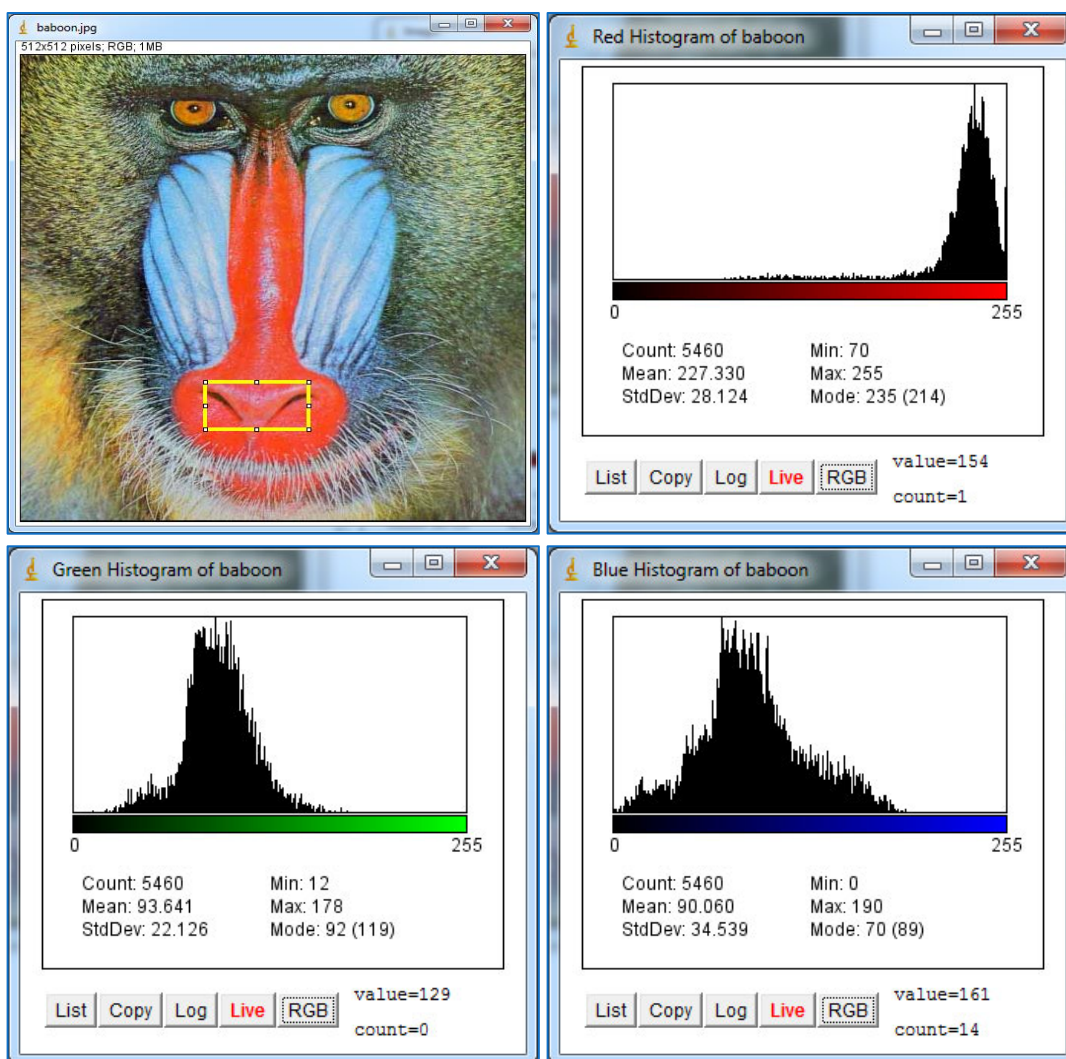


Figura 4.10.2. De izquierda a derecha y de arriba abajo: imagen con una selección de área sobre una zona con una gran carga de rojo, histograma correspondiente al canal rojo, histograma correspondiente al canal verde, histograma correspondiente al canal azul

Con la ventana del histograma activa, es posible acceder al menú de la Figura 4.10.3 a través de la combinación de teclas Alt + H. Este menú permite configurar los últimos parámetros del histograma: *Bins*, permite especificar el número de columnas manualmente; *Use pixel value range*, determina automáticamente el rango de valores contenidos en el eje de abscisas, que van desde el mínimo nivel de brillo encontrado en la imagen hasta el máximo; *X Min* y *X Max* permiten determinar manualmente el rango de valores

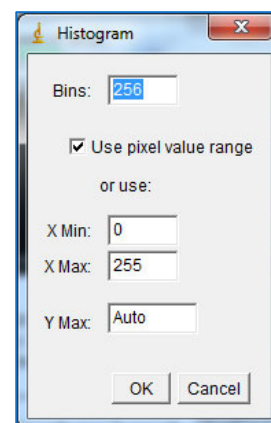


Figura 4.10.3. Menú emergente Histogram

contenidos en el eje de abscisas; *Y Max* permite determinar el valor máximo que se alcanza en el eje de ordenadas (introduciendo la cadena *Auto* se hace coincidir el máximo con la altura de la columna más alta).

4.11. Plot Profile

La herramienta *Plot Profile* (trazar perfil) genera un gráfico bidimensional en el que se representa nivel de brillo frente a distancia y que contiene una curva que informa del nivel de brillo que encuentra una selección lineal previamente realizada en su camino. Esto se aprecia en la Figura 4.11.1, en la que aparece el perfil de una selección lineal que atraviesa áreas claras uniformes intercaladas con manchas oscuras, y que por tanto se compone de zonas relativamente planas y valles.

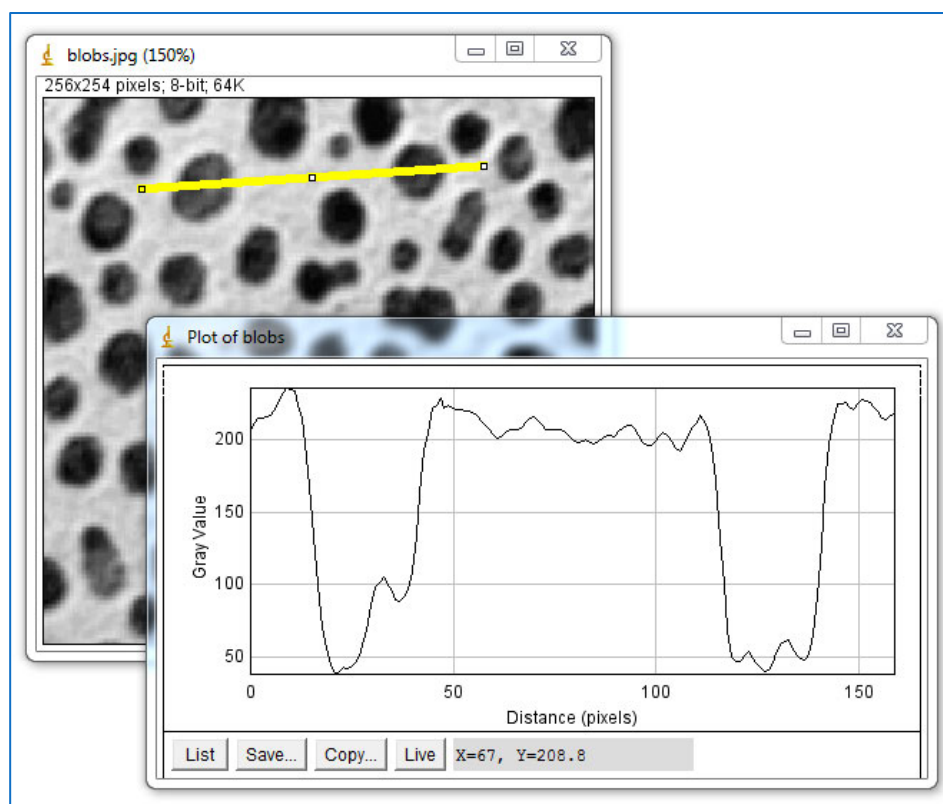


Figura 4.11.1. Selección lineal junto a su perfil

También es posible obtener el perfil de selecciones de área (o selecciones lineales con grosor), en cuyo caso se hace corresponder a cada distancia el valor medio de los niveles de brillo de los píxeles que abarque la selección de área para dicha distancia; es decir, la anchura de la selección se toma como la distancia total y, para cada distancia, se hace la media de un número de valores que coincide con la altura de la selección (estos roles se pueden invertir marcando la casilla *Vertical Profile* en la ventana *Profile Plot Options*, submenú *Options*, menú *Edit*).

La ventana que contiene el perfil presenta además botones que implementan funcionalidades que ya se han comentado en apartados anteriores: *List*, *Save*, *Copy* y *Live*. A su derecha hay un campo de texto que proporciona las coordenadas XY de la curva para explorarla más detalladamente al desplazar el cursor sobre ella.

Para terminar hay que destacar que es posible representar más de un perfil en el mismo gráfico, pero para ello hay que usar la utilidad *Multi Plot* del Gestor de Regiones de Interés, *ROI Manager* (ver el apartado 4.14 Submenú *Tools*).

4.12. Surface Plot

Esta herramienta se puede considerar una evolución de la herramienta anterior, *Plot Profile*, como se observa en la Figura 4.12.1:

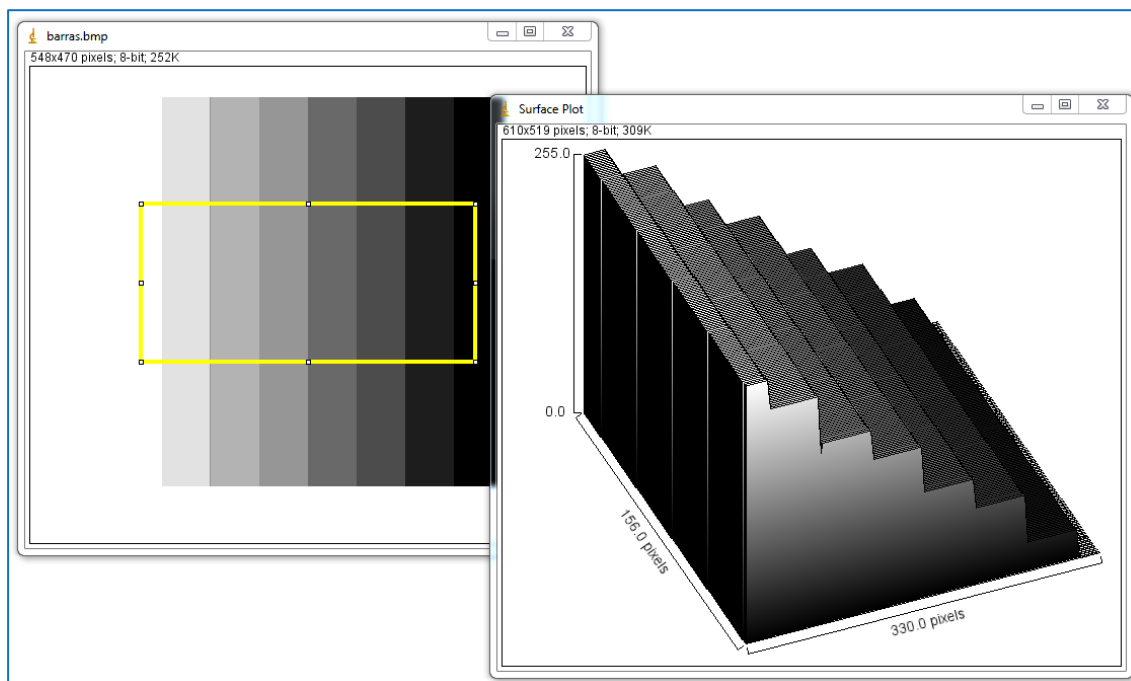


Figura 4.12.1. Selección de área sobre una carta de barras en escala de grises junto al correspondiente gráfico de superficie con forma escalonada

Surface Plot genera un gráfico tridimensional que contiene una superficie formada por tantos puntos como píxeles contenga la selección de área previamente hecha (o, en su defecto, toda la imagen activa), de forma que las coordenadas XY de los puntos son las coordenadas XY de los píxeles y la coordenada Z de los puntos coincide con el nivel de brillo de los píxeles. Se puede considerar esta representación tridimensional como la agrupación de tantos perfiles bidimensionales (obtenidos a través de *Plot Profile*) como píxeles de altura tenga la selección o imagen (de hecho esta filosofía es el fundamento de esta herramienta, como se verá más adelante).

Al aplicar esta herramienta sobre una pila de imágenes se obtiene una nueva pila con tantos gráficos tridimensionales como capas tenía la pila original.

Antes de generarse el gráfico con la superficie se presenta al usuario el menú *Surface Plotter* (ver Figura 4.12.2) para configurar la operación. Permite ajustar los siguientes parámetros:

- *Polygon Multiplier*. Permite ajustar manualmente el número de perfiles bidimensionales usados para construir la superficie. Puede tomar valores desde 10% a 200%, y cuanto mayor sea este valor mayor continuidad tendrá la superficie, aunque también supondrá una mayor carga computacional.
- *Draw Wireframe*. Marcando esta casilla se delinea el contorno de cada perfil utilizado para construir la superficie (se utiliza el algoritmo de renderización *Wireframe*). El gráfico de la Figura 4.12.1 se construyó marcando esta opción.
- *Shade*. Al marcar esta casilla se sombrea la superficie de forma que, normalmente, las zonas correspondientes a píxeles más oscuros reciben un sombreado más oscuro, lo cual contribuye a la que la percepción tridimensional de la superficie sea más cómoda. El gráfico de la Figura 4.12.1 se construyó marcando esta opción. Para el sombreado se tiene en cuenta la LUT, lo cual resulta muy útil en el caso de las imágenes indexadas, puesto que la superficie se sombrea con los colores correspondientes a los índices de los píxeles asociados a cada zona.
- *Draw Axis*. Marcando esta opción se dibujan y etiquetan los tres ejes.
- *Source Background is Lighter*. Al marcar esta casilla se considera que el fondo de la imagen es la parte clara, por lo que las zonas con bajo nivel de brillo pasan a representarse con puntos elevados y viceversa.
- *Fill Plot Background with Black*. Marcando esta opción se utiliza un fondo negro para el gráfico (en caso contrario se utiliza un fondo blanco).

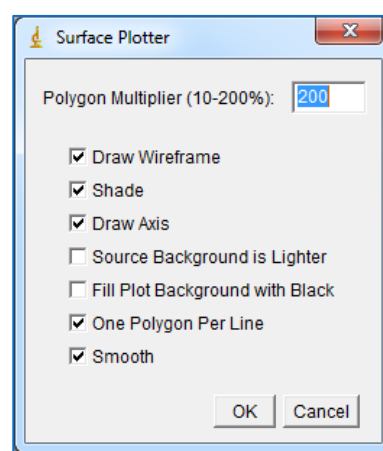


Figura 4.12.2. Menú *Surface Plotter*

- *One Polygon Per Line*. Al marcar esta opción, para construir la superficie se utilizan tantos perfiles como líneas de altura tenga la selección o imagen, independientemente del valor indicado en el campo *Polygon Multiplier*.
- *Smooth*. Esta opción suaviza los cambios bruscos y bordes abruptos de la superficie.

El *plugin Interactive 3D Surface Plot*, presente por defecto en la versión 1.48q de ImageJ (submenú *3D*, menú *Plugins*), puede entenderse como una versión mejorada con funcionalidades extendidas de la herramienta *Surface Plot*.

4.13. Submenú Gels

En el submenú *Gels* se aglutinan todas las herramientas relacionadas con el análisis de geles electroforéticos. Estos son el resultado de aplicar las diferentes técnicas agrupadas bajo el nombre de Electroforesis en Gel. Estas técnicas están orientadas a separar moléculas para su posterior análisis en base a propiedades como el tamaño, la forma o el punto isoelectrico. La electroforesis en gel se basa en aplicar una diferencia de potencial a una matriz sólida y porosa hecha de gel (e.g., polímeros entrelazados) sobre la que se sitúan las moléculas de interés, de modo que éstas son afectadas por una fuerza electromotriz que las desplaza a diferentes velocidades hacia el cátodo o el ánodo. Estas velocidades pueden ser, por ejemplo, inversamente proporcionales al tamaño (en los ácidos nucleicos), lo cual permite separar en el gel las moléculas en base a esta propiedad. Una vez se han separado, se pueden revelar, dependiendo del caso, con colorantes y tinciones, luz ultravioleta o autorradiografías. Es posible conseguir separaciones simultáneas introduciendo varias mezclas, lo cual da lugar a “calles” o separaciones paralelas, en las que se encontrarán distintas bandas para los componentes de cada mezcla. Si dos bandas de calles contiguas se encuentran a la misma altura se puede deducir que las mezclas correspondientes contienen moléculas que han atravesado el gel a la misma velocidad. Teniendo en cuenta esto último, y haciendo electroforesis en gel de mezclas desconocidas y de marcadores (mezclas que contienen moléculas de propiedades conocidas) es posible estimar las propiedades de las moléculas de la mezcla desconocida.

Los geles electroforéticos se pueden escanear para ser estudiados con herramientas de tratamiento digital de imagen como ImageJ teniendo en cuenta que la densidad óptica es proporcional a la concentración de la muestra. Pero para que los resultados en el dominio digital sean fidedignos, el escáner ha de cumplir los siguientes requisitos⁷:

⁷ Para más información, consultar: Gassman et al. “*Quantifying Western blots: pitfalls of densitometry*” [38]

- Las muestras se deben escanear en “modo transmisión”: la fuente de luz debe situarse en el lado del escáner opuesto al del detector, con la muestra entre ambos (es decir, la luz emitida atraviesa el gel e incide en el detector). Los escáneres que operan en “modo reflexión” no sirven. En estos escáneres, la fuente de luz y el detector se localizan en el mismo lado, de modo que la luz emitida se refleja en la muestra e incide en el detector. En esta reflexión se dan fenómenos que complican el proceso de tal forma que no es aplicable a la digitalización de geles electroforéticos (e.g., reflexiones dobles).
- El escáner debe presentar una respuesta de intensidad lineal.
- Es conveniente que la imagen digitalizada tenga una alta resolución y una alta profundidad de bit.

A continuación se muestra un ejemplo de análisis de un gel electroforético con ImageJ en el que se analiza la concentración de tres bandas que están a la misma altura en tres calles paralelas. Se parte del gel que se muestra en la Figura 4.13.1. En primer lugar hay que configurar el proceso a través de la ventana emergente que también se muestra en la Figura 4.13.1, a la cual se accede haciendo *click* en *Gel Analyzer Options* en el submenú *Gels*. En ella se pueden configurar los siguientes parámetros:

- *Vertical/Horizontal scale factor*. En estos campos de texto se puede especificar si se escalan (y en tal caso, en qué proporción) los perfiles de las calles que se muestran tras pulsar *Plot Lanes*, tanto verticalmente como horizontalmente. En este caso se dejan los valores por defecto: 1 y 1 (es decir, no se escalan los perfiles).
- *Uncalibrated OD*. Densidad óptica sin calibrar. Al marcar esta casilla, ImageJ convierte el nivel de brillo de los píxeles a valores de densidad óptica directamente en base a la siguiente fórmula (sin calibración):

$$Unc. O. D. = \log_{10} \left(\frac{255}{nivel\ de\ brillo} \right) \quad (4.13.13)$$

Esta ecuación sólo es aplicable a imágenes de tipo *8-bit*. En este caso, se marca para evitar el paso de la calibración (ver el apartado 4.9 *Calibrate*).

- *Label with percentages*. Si se marca esta casilla, al usar la utilidad *Label Peaks* (etiquetar picos), que se comenta más adelante, se calcula el porcentaje que supone el área de cada pico con respecto al total (la suma de las áreas de todos los picos) y se usa para etiquetar los picos. Además, los porcentajes se trasladan a la tabla de resultados. En este caso se marca.
- *Invert peaks*. Al marcar esta opción se invierten los picos: las zonas más oscuras que el fondo suponen picos positivos y viceversa. En este caso, se marca.

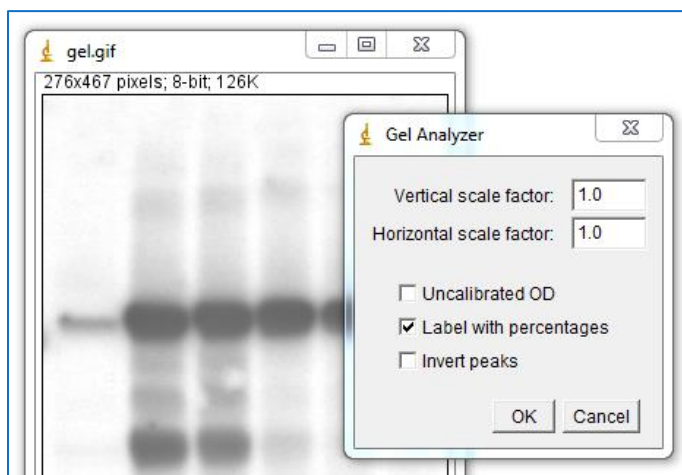


Figura 4.13.1. Imagen de un gel electroforético junto al menú *Gel Analyzer*

Una vez configurada la operación, se pasa a usar la herramienta de selección rectangular para seleccionar la parte de cada calle que contiene la banda de interés. Una vez hecho esto, se pulsa en *Select First Lane*, en el submenú *Gels*. Se asigna automáticamente el índice 1 a esta selección. Se asume que las calles son verticales a no ser que el ancho de la selección sea al menos el doble de su altura. A

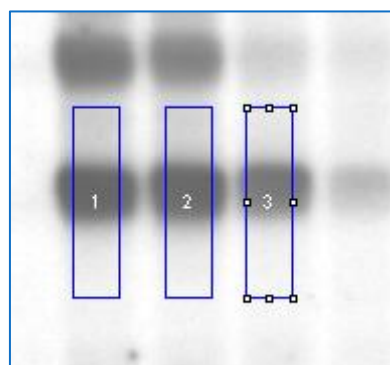


Figura 4.13.2. Selecciones rectangulares sobre las bandas de interés

continuación se utiliza la misma selección, arrastrándola con el ratón, para abarcar la banda contigua en la calle adyacente y se pulsa en *Select Next Lane*, en el submenú *Gels*. Aunque manualmente no se haya colocado la selección rectangular a la misma altura que la anterior, al pulsar en *Select Next Lane* se recoloca de forma automática. Se repite el proceso con todas las calles de interés (en este caso, sólo una más). El resultado se muestra en la Figura 4.13.2.

El siguiente paso es obtener los perfiles pulsando en *Plot Lanes*. Esto es equivalente a utilizar la herramienta *Plot Profile* para cada selección y presentar los resultados en la misma ventana. A continuación hay que delimitar el área correspondiente a cada pico de interés para medirlas. Para ello se dibujan los límites con la herramienta de selección lineal recta (*Straight Line Selection Tool*), como se muestra en la Figura 4.13.3. Sobre las áreas delimitadas se aplica la herramienta de selección *Wand Tool*, que selecciona automáticamente toda el área de cada pico. Cada vez que se aplica la herramienta *Wand Tool* sobre el área de un pico, se mide y su valor se traslada a la tabla de resultados. Esto último se muestra en la Figura 4.13.4.

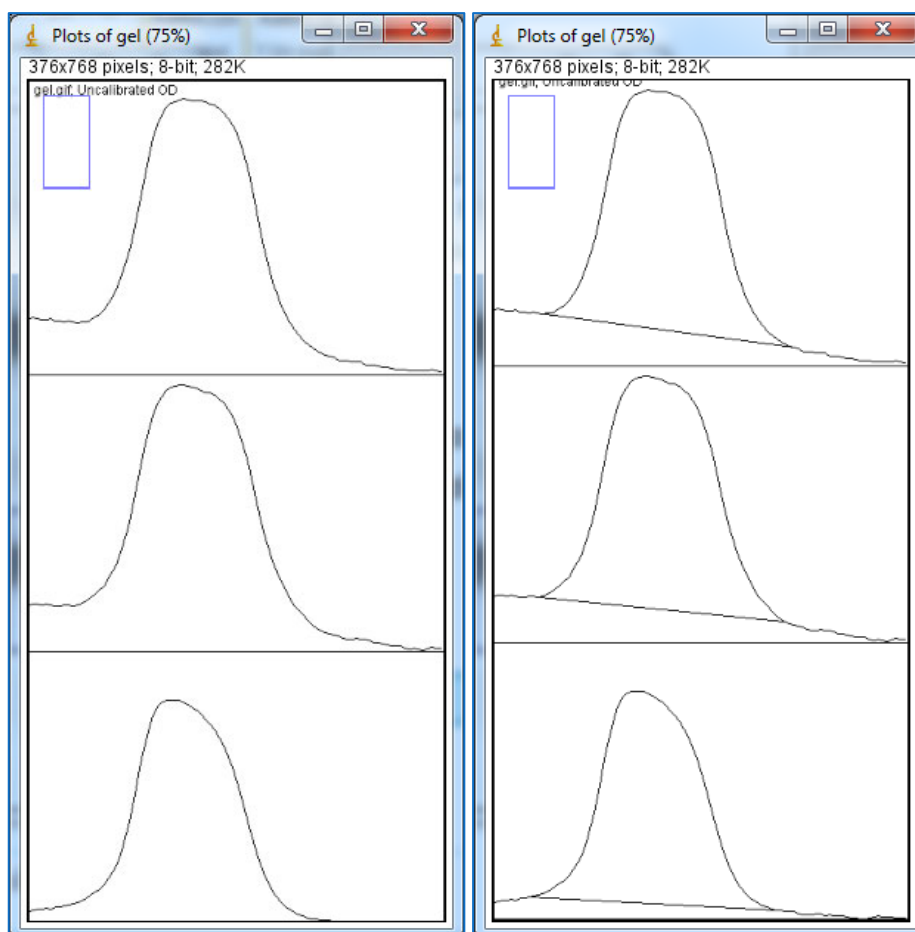


Figura 4.13.3. Izquierda: perfiles de las calles (picos de densidad óptica de las bandas de interés), uno encima de otro. Derecha: áreas de los picos de densidad óptica de las bandas de interés delimitadas con la herramienta de selección lineal recta

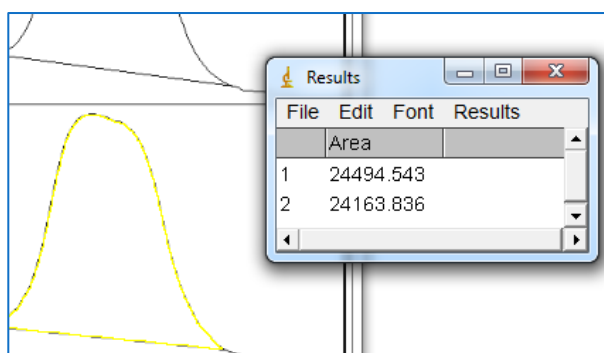


Figura 4.13.4. Área de un pico seleccionada junto a la tabla de resultados, que contiene el tamaño de los dos primeros picos

Se repite el proceso para todos los picos y al terminar se obtiene toda la información en la tabla de resultados. Para obtener el porcentaje que supone el tamaño de cada pico con respecto del total y utilizar además este dato como etiqueta del pico basta con pulsar en *Label Peaks*, submenú *Gels*. El resultado se muestra en la Figura 4.13.5:

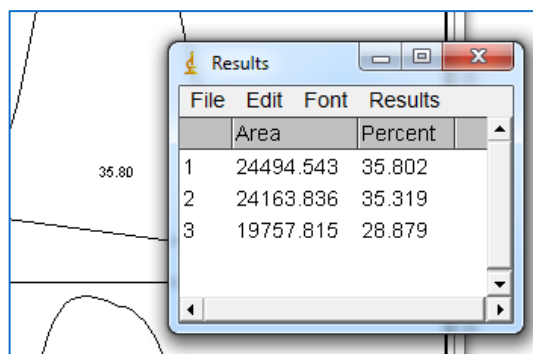


Figura 4.13.5. Tabla de resultados con todas las áreas y los porcentajes que suponen cada una con respecto al total

De esta forma se puede evaluar indirectamente la concentración en las bandas de las calles de un gel electroforético.

Las funciones *Reset* y *Re-plot lanes*, que se incluyen en el submenú *Gels* pero que no se han comentado hasta ahora, sirven, respectivamente, para reemprender el proceso desde el principio (eliminar las selecciones y vaciar el panel de perfiles y la tabla de resultados) y para obtener por segunda vez los perfiles sin tener que empezar desde cero.

4.14. Submenú Tools

En este submenú se agrupan varias herramientas adicionales orientadas al análisis estadístico de las imágenes.

4.14.1. Save XY Coordinates

Como se explica en el propio panel emergente, la herramienta *Save XY Coordinates* permite exportar a un fichero de texto las coordenadas y el nivel de brillo de todos los píxeles que no formen parte del fondo, entendiendo por fondo el conjunto de todo los píxeles cuyo nivel de brillo sea el del primer píxel de la imagen (el de la esquina superior izquierda). El valor de este nivel de brillo se muestra en el campo *Background value* y se puede cambiar manualmente. Al pulsar *OK* se despliega una ventana que permite elegir la ubicación del fichero de texto y se guarda. Como se aprecia en la parte derecha de la Figura 4.14.1, el fichero de texto exportado consta de tres columnas: la coordenada X (de izquierda a derecha), la coordenada Y (de abajo a arriba; esto se puede invertir marcando la casilla *Invert y coordinates off*) y el nivel de brillo. En el caso de las imágenes en color, el fichero de texto contiene cinco columnas: la coordenada X, la coordenada Y, el nivel de la componente R, el nivel de la componente G y el nivel de la componente B.

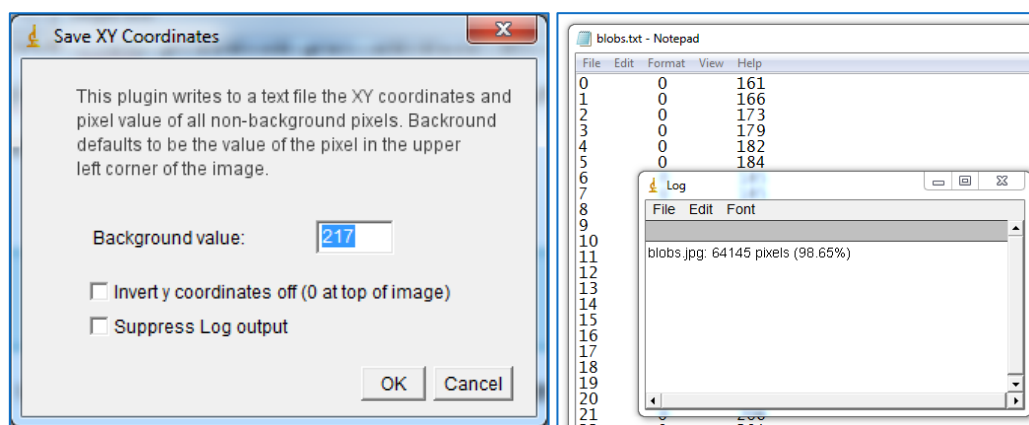


Figura 4.14.1. Izquierda: menú emergente *Save XY Coordinates*. Derecha: fichero de texto con las coordenadas y el nivel de brillo de los píxeles que no forman parte del fondo

Al terminar de exportar los datos al fichero de texto se abre la ventana *Log*, que muestra el número de píxeles incluidos en el fichero y el porcentaje que estos suponen sobre el total de píxeles de la imagen. Se puede evitar que se despliegue esta ventana marcando la casilla *Suppress Log output*.

4.14.2. Fractal Box Count

Esta herramienta permite estimar la Dimensión Fractal D de una imagen binaria, la cual se puede tomar como un descriptor de complejidad de patrones (texturas, formas de células, vascularización,...) cuando medidas como la longitud o el diámetro no resultan ser descriptores adecuados. Esto sucede con formas fractales. Las formas fractales están constituidas por elementos cada vez más pequeños, como sucede en la Figura 4.14.2, en la que se presenta la Curva de Koch. Cada evolución de la Curva de Koch aumenta en un tercio su longitud a base de incorporar versiones más reducidas de su elemento fundamental. A medida que el fractal evoluciona aumenta su longitud y al mismo tiempo aumenta la dificultad para medirla, puesto que los elementos son cada vez más pequeños. Así, llegará un punto en que los elementos sean aún más reducidos que la precisión del instrumento de medida. Esto último hace que las medidas euclídeas clásicas resulten inadecuadas y fuerza la aparición de la Geometría Fractal y el concepto de Dimensión Fractal, que es una generalización de la Dimensión Euclídea. Se dice que la Dimensión Fractal de un objeto geométrico es D si se cumple lo siguiente:

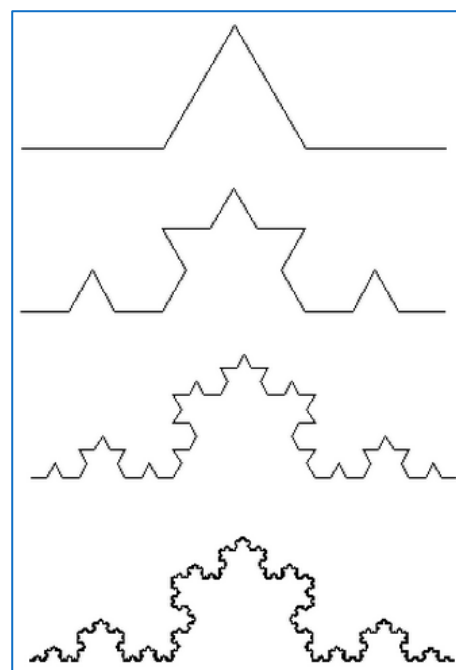


Figura 4.14.2. Forma fractal (Curva de Koch)

$$N(L)L^D = 1 \quad (4.14.14)$$

donde $N(L)$ es el número de objetos elementales de tamaño L que recubren completamente el objeto. Despejando D en la expresión 4.14.13 resulta

$$D = \log_L \left(\frac{1}{N(L)} \right) = -\log_L(N(L)) = -\frac{\log(N(L))}{\log(L)} \quad (4.14.15)$$

En el caso de la Curva de Koch se necesitan cuatro elementos de un tercio de unidad, es decir, $N(L) = 4$ y $L = 1/3$, así que

$$D = -\frac{\log(4)}{\log\left(\frac{1}{3}\right)} \approx 1,26 \quad (4.14.16)$$

Es decir, tiene una dimensión mayor que 1 y menor que 2 (entre la línea y el plano). Como se ha explicado, la Dimensión Fractal es una generalización de la Dimensión Euclídea, por lo que debería ser aplicable a formas clásicas. Por ejemplo, la Figura 4.14.3 presenta un cuadrado cuya superficie se supone la unidad subdividido según elementos cuadrados de diferente superficie (L^2) que siempre lo recubren completamente.

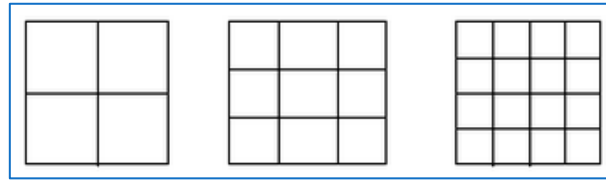


Figura 4.14.3. Cuadrado de superficie la unidad subdividido según elementos de diferente tamaño

En el primer caso, $N(L) = 4$ y $L = 1/2$. En el segundo caso, $N(L) = 9$ y $L = 1/3$. En el tercer caso, $N(L) = 16$ y $L = 1/4$. Por tanto,

$$D = -\frac{\log(4)}{\log\left(\frac{1}{2}\right)} = 2 \quad (4.14.17)$$

$$D = -\frac{\log(9)}{\log\left(\frac{1}{3}\right)} = 2 \quad (4.14.18)$$

$$D = -\frac{\log(16)}{\log\left(\frac{1}{4}\right)} = 2 \quad (4.14.19)$$

Es decir, el cuadrado tiene dimensión 2, lo cual concuerda con la Geometría Euclídea.

Fractal Box Count se basa en contar el número de cajas de tamaño creciente (los cuales se pueden determinar manualmente al principio de la operación) que son necesarias para cubrir el contorno de un objeto de una imagen binaria según el método descrito en T. G. Smith, Jr., G. D. Lange and W. B. Marks, *Fractal Methods and Results in Cellular Morphology*, *J Neurosci Methods*, 69:1123-126,1996. Al concluir la operación, se

despliega una ventana que contiene un gráfico de ejes logarítmicos en el que se representa $N(L)$ (*count*, conteo del número de cajas necesarias) frente a L (*box size*, tamaño de la caja). En este gráfico se representan los datos resultantes de la operación y se acomoda una recta sobre ellos. Teniendo en cuenta esto y la definición (4.14.15) se deduce que la Dimensión Fractal D coincide con el opuesto de la pendiente de dicha recta: $-\log(N(L))/\log(L)$. Este valor se presenta en la parte superior derecha del gráfico. Además, todos los datos *box size* y *count* utilizados en la operación se almacenan en la tabla de resultados. La ventana que contiene el gráfico también alberga botones que ofrecen funcionalidades adicionales que ya se han comentado en apartados anteriores: *List*, *Save* y *Copy*.

A continuación se muestra el resultado de aplicar la herramienta *Fractal Box Count* a una imagen binaria que contiene un cuadrado. Como era de esperar, la estimación de su dimensión fractal se acerca a 2: 1,937.

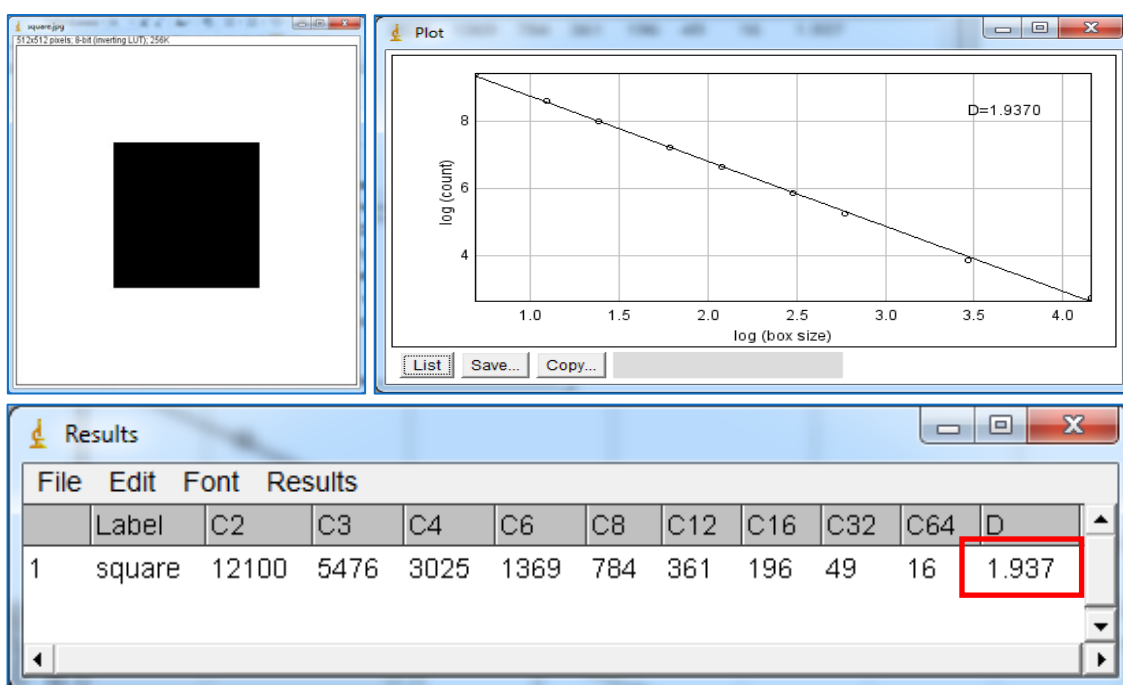


Figura 4.14.4. Arriba, izquierda: imagen binaria de un cuadrado. Arriba, derecha: gráfico generado por *Fractal Box Count* para la imagen anterior. Abajo: información generada por *Fractal Box Count* para la imagen anterior (en rojo se destaca la estimación de su dimensión fractal: 1,937)

Se repite el ejemplo anterior para el caso de una línea recta. En este caso se espera obtener una estimación de la dimensión fractal cercana a 1:

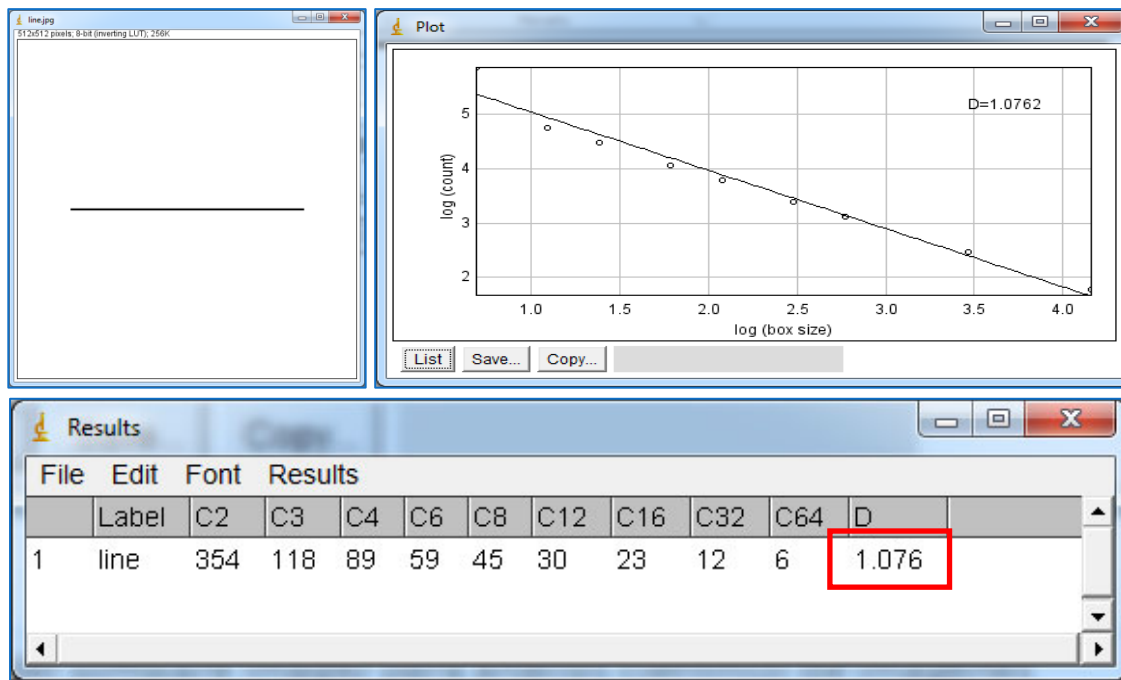


Figura 4.14.5. Arriba, izquierda: imagen binaria de una línea. Arriba, derecha: gráfico generado por *Fractal Box Count* para la imagen anterior. Abajo: información generada por *Fractal Box Count* para la imagen anterior (en rojo se destaca la estimación de su dimensión fractal: 1,076)

Para terminar se repite una vez más el ejemplo, en este caso para la Curva de Koch, por lo que se espera obtener una estimación de dimensión fractal cercana a 1,26:

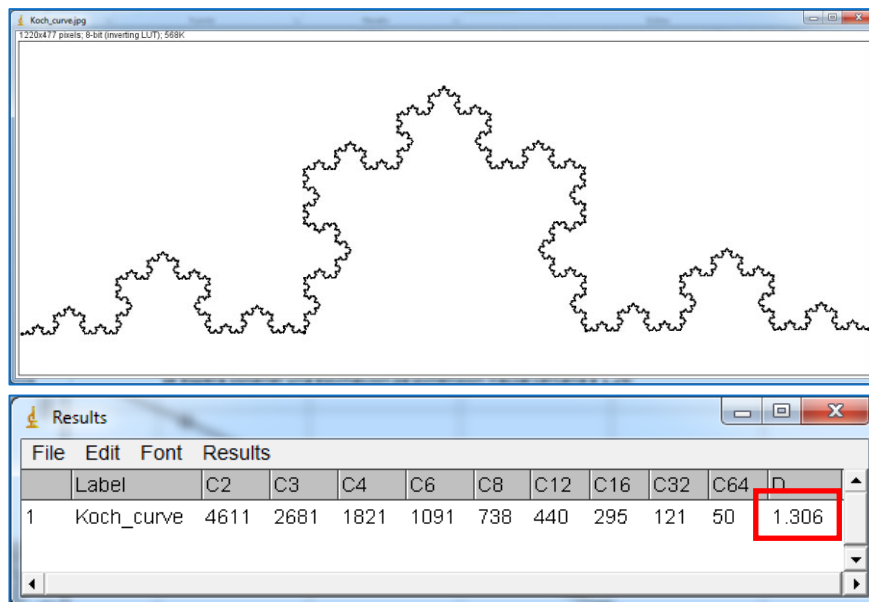


Figura 4.14.6. Arriba: imagen binaria de una Curva de Koch. Abajo: información generada por *Fractal Box Count* para la imagen anterior (en rojo se destaca la estimación de su dimensión fractal: 1,306)

4.14.3. Analyze Line Graph

Esta herramienta permite extraer la información contenida en las curvas de imágenes de gráficos bidimensionales. Se basa en recuperar las coordenadas de los puntos que componen la curva empleando a su vez la herramienta *Analyze Line Graph*, la cual ya se ha comentado. Para emplear esta herramienta es necesario seguir una serie de pasos previos.

- En primer lugar es necesario abrir la imagen y empezar a aislar la curva (aproximadamente). Para ello se pueden utilizar las herramientas de selección de área o las herramientas de dibujo. El algoritmo asume que el fondo del gráfico es blanco, así que, en caso contrario, hay que emplear la herramienta *Invert*, menú *Edit*.

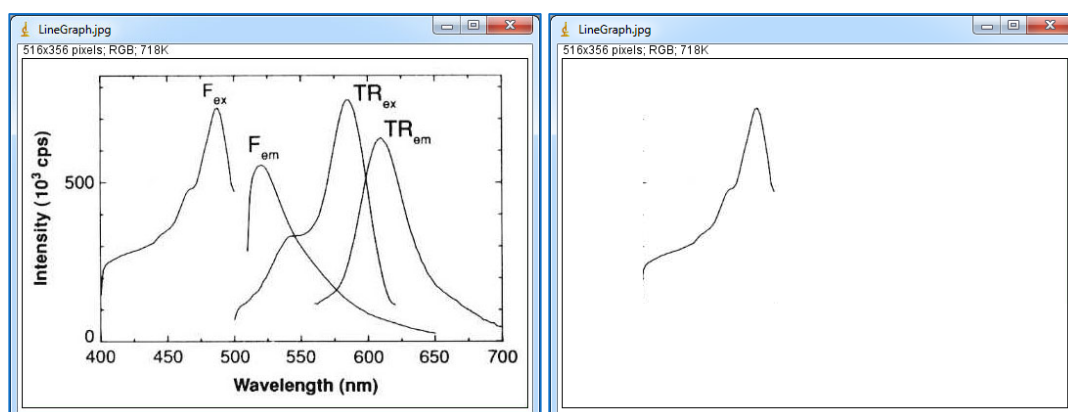


Figura 4.14.7. Izquierda: imagen de un gráfico bidimensional con varias curvas. Derecha: curva de interés aislada

- A continuación se ha de umbralizar la imagen de forma que se extraiga la curva de interés. Para ello se puede emplear la herramienta de umbralización (*Threshold*, submenú *Adjust*, menú *Image*):

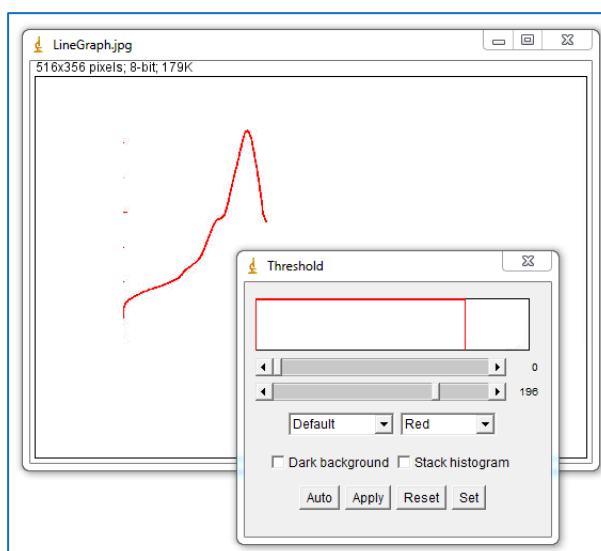


Figura 4.14.8. Umbralización de la curva de interés

- En este punto la curva se ha extraído, pero el proceso no ha sido perfecto: aún quedan en la imagen objetos ajenos a la curva de interés. El algoritmo requiere que el aislamiento sea perfecto para obtener

resultados precisos. Para asegurar esto, se selecciona la curva con la herramienta de selección *Wand Tool* (situando el cursor sobre la curva y haciendo *click*) y se hace uso de la herramienta *Clear Outside* (limpiar afuera), en el menú *Edit*. Después de esto se puede tener la certeza de que en la imagen no hay más objetos que la curva:

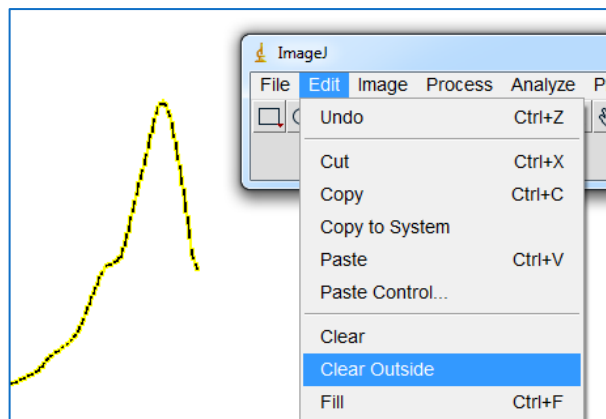


Figura 4.14.9. Curva de interés seleccionada con la herramienta *Wand Tool* y herramienta *Clear Outside*

- En este punto de la operación se usa la herramienta *Analyze Line Graph*, lo cual despliega la ventana emergente *Line Graph*. En esta ventana se muestra la curva procesada en un gráfico bidimensional que enfrenta altura en píxeles (*Y pixels*) frente a anchura en píxeles (*X pixels*). Si la imagen hubiera estado calibrada espacialmente, se habrían empleado las unidades oportunas. Además se ofrecen los botones *List*, *Save* y *Copy*, que permiten listar en la tabla de resultados las coordenadas de los puntos que componen la curva, guardar la información en un archivo de texto y trasladarla al portapapeles del sistema respectivamente. Esto se muestra en la Figura 4.14.10:

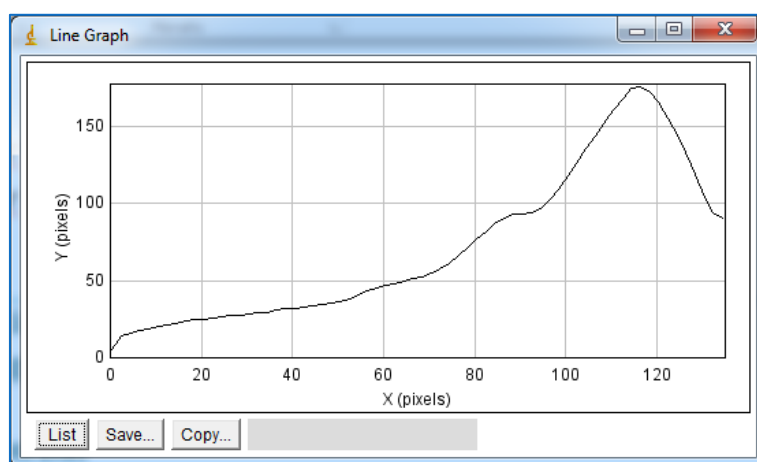


Figura 4.14.10. Ventana *Line Graph*

En la Figura 4.14.11 se muestra una hoja de cálculo a la que se han trasladado los datos de la curva de la Figura 4.14.10 a través del botón *Copy*.

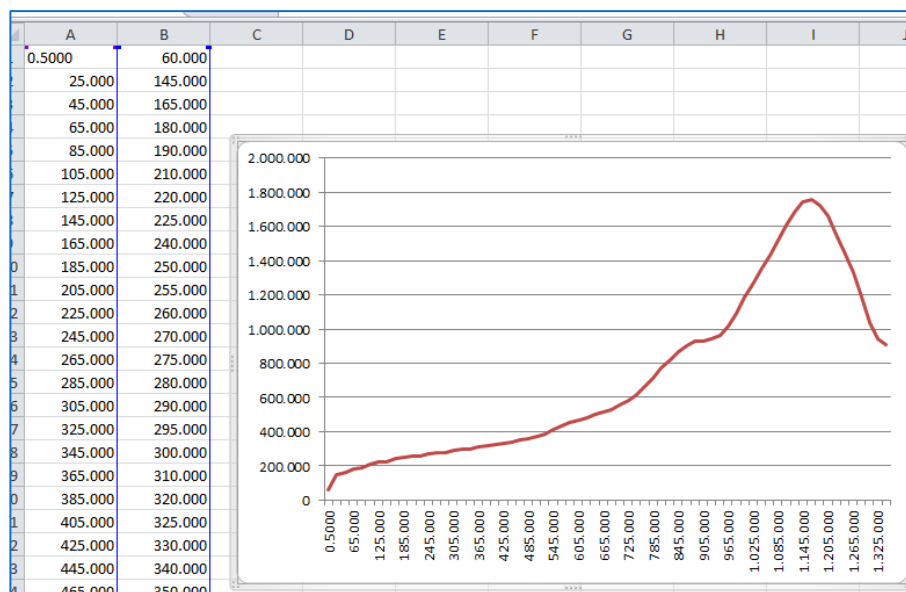


Figura 4.14.11. Hoja de cálculo con las coordenadas de los puntos que conforman la curva de interés

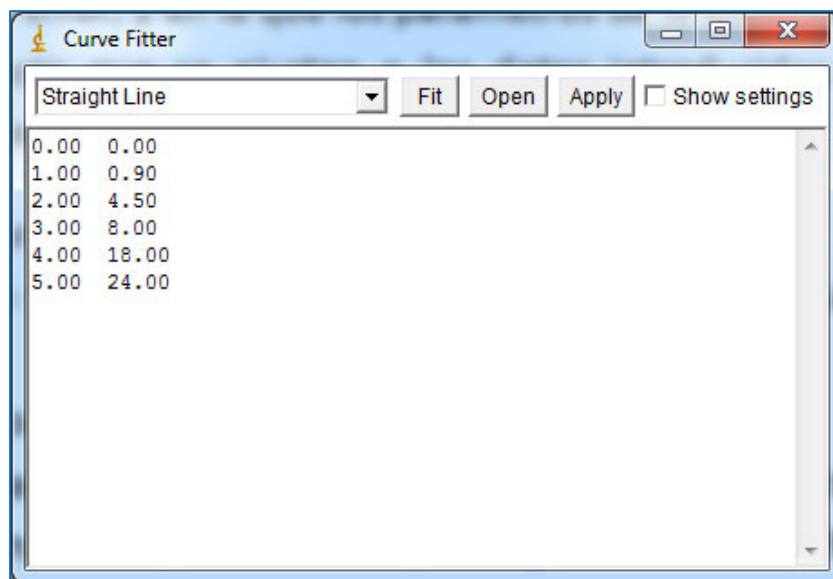
4.14.4. Curve Fitting

Curve Fitting es la herramienta de ajuste de curvas de ImageJ. Se basa en un algoritmo iterativo que sigue una estrategia de ensayo y error, y en la que los parámetros de la ecuación en cuestión se reajustan una y otra vez hasta que se ajustan a los datos introducidos con suficiente precisión. Las fases fundamentales del algoritmo son las siguientes:

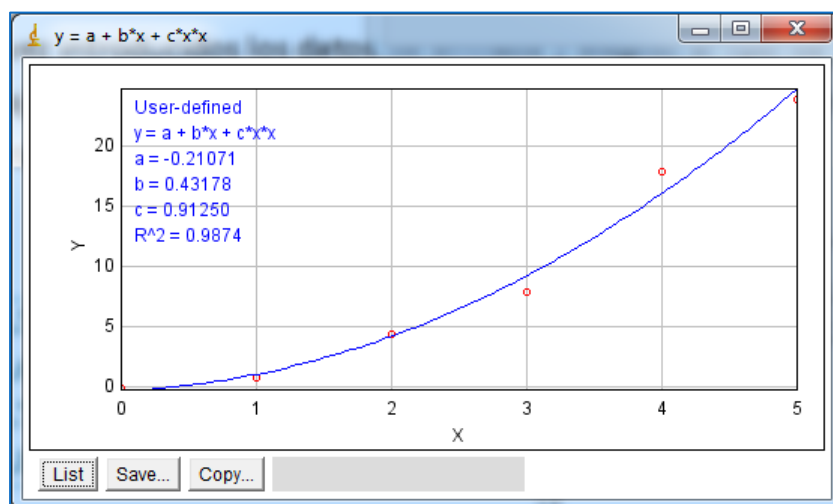
- 1. Se hacen las primeras conjeturas sobre todos los parámetros no lineales
- 2. Se computa el modelo, se compara con el juego de datos introducidos y se calcula el error cometido
- 3. Si el error es mayor de lo permitido, se vuelve a la fase 2.

Evidentemente el algoritmo sólo finaliza cuando el error cometido es tolerable, lo cual no tiene por qué llegar a suceder (en casos muy complicados). Para evitar que caiga en un bucle infinito se impone un número de iteraciones máximo.

La herramienta ofrece una interfaz como la de la Figura 4.14.12. Contiene una lista desplegable que permite elegir el tipo de función que se tiene que ajustar a los datos de entrada de entre un gran número de tipos de funciones: *Straight Line* (lineal), *nth Degree Polynomial* (polinómica de enésimo grado), *Power* (potencia), *Exponential* (exponencial),... También es posible definir manualmente la función eligiendo *User-defined*, aunque el número de parámetros se limita a seis y además se empeora el rendimiento. Bajo la lista desplegable hay un área de texto para introducir los datos tabulados (en columnas o separados por espacios, coma o punto y coma). Los datos de entrada también se pueden introducir a través de un archivo de texto con los datos tabulados pulsando el botón *Open* y navegando hasta dicho archivo.

Figura 4.14.12. Interfaz *Curve Fitter*

Una vez introducidos los datos de entrada y elegido el tipo de función se puede pulsar el botón *Fit* (ajustar), lo cual genera un gráfico con los datos de entrada y la curva correspondiente ajustada, la cual se describe completamente, tal y como muestra la Figura 4.14.13. En caso de que se haya elegido *User-defined* en la lista desplegable, antes de generarse el gráfico se pide al usuario que defina la función y especifique sus parámetros, así como el número máximo de iteraciones, los reinicios⁸ y el error permitidos.

Figura 4.14.13. Curva ajustada a los datos de entrada con la herramienta *Curve Fitting*

Si en la interfaz se marca previamente la casilla *Show settings*, además del gráfico se despliega la ventana *Log* con información detallada acerca del proceso: número de iteraciones, tiempo empleado,... Si el resultado es satisfactorio, se puede guardar el gráfico como una imagen de tipo *32-bit* pulsando el botón *Apply* en la

⁸ El algoritmo puede reiniciarse para trabajar desde diferentes puntos de partida y buscar el mejor resultado

interfaz. También es posible almacenar la información de forma numérica a través de los botones *List*, *Save* y *Copy* que aparecen bajo el gráfico y cuyas funciones ya se han discutido.

4.14.5. ROI Manager

La herramienta *ROI Manager* o Gestor de Regiones de Interés permite trabajar a la vez con múltiples selecciones de cualquier tipo (de área, lineales, puntuales, textuales,...) y de cualquier origen (diferentes regiones de una misma imagen, diferentes capas de una misma pila, diferentes imágenes o pilas,...)⁹. Para ello presenta una interfaz (ver Figura 4.14.14) desde la que se puede acceder a múltiples funcionalidades.

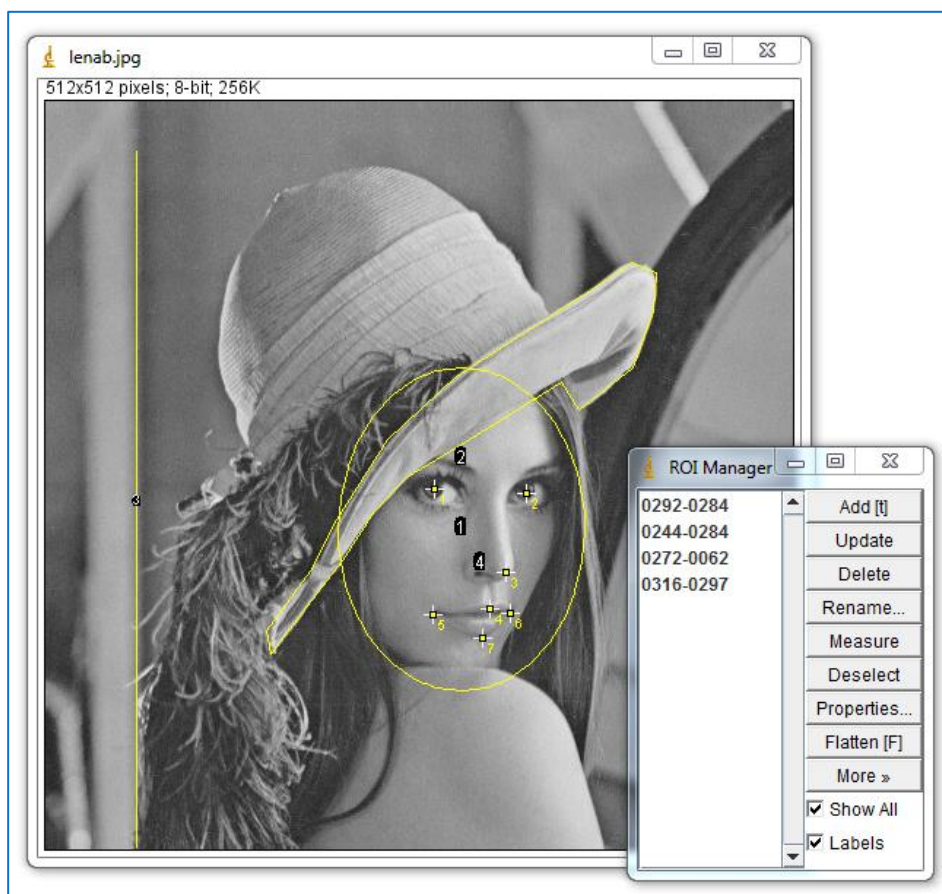


Figura 4.14.14. Imagen con múltiples selecciones de distintos tipos junto al Gestor de Regiones de Interés

- Lista de selecciones. Abarca la mitad izquierda de la interfaz. Si la selección se ha renombrado con la herramienta *Rename*, aparece representada por el nombre elegido. De lo contrario se usa el nombre por defecto, que para las selecciones hechas sobre imágenes individuales consta de dos partes separadas por un guion: la primera, la coordenada Y del centro de la selección, y la segunda, la coordenada X. Para las selecciones hechas sobre pilas, hay una tercera parte, el número de capa, que aparece al principio del nombre separado por otro guion. Al seleccionar una entrada de la lista se representa la selección correspondiente de forma no invasiva sobre la imagen activa (siempre que sea posible), sea o no su

⁹ No confundir con los recubrimientos de las imágenes (ver el apartado *Overlay*)

origen. Además, si se elige una de las funcionalidades listadas en la parte derecha de la interfaz se aplica a la selección elegida. Como en cualquier otra lista, es posible seleccionar múltiples entradas contiguas manteniendo presionada la tecla *Shift*, o múltiples entradas no contiguas presionando de forma continua la tecla *Ctrl*. Para no seleccionar ninguna entrada basta con presionar el botón *Deselect*.

- *Add*. Permite añadir la selección actual al gestor. Se añade en la lista de selecciones a continuación de la última selección.
- *Update*. Permite actualizar o reemplazar la selección elegida en la lista de selecciones del gestor con la selección actual.
- *Delete*. Permite eliminar del gestor la selección o selecciones elegidas en la lista. Si no hay ninguna entrada seleccionada en la lista, se eliminan todas las selecciones (previa confirmación).
- *Rename*. Permite renombrar la selección elegida en la lista (sólo una cada vez).
- *Measure*. Aplica la herramienta *Measure*, comentada en apartados anteriores, a la selección o selecciones elegidas en la lista del gestor, tras lo cual se despliega la ventana de resultados. En caso de que no haya ninguna entrada de la lista seleccionada, se miden todas las selecciones.
- *Deselect*. Deselecciona cualquier entrada de la lista previamente seleccionada. Tras esto, la mayoría de las funcionalidades del gestor se aplican a todas las selecciones almacenadas.
- *Properties*. Despliega un menú emergente en el que es posible reconfigurar varias propiedades de las selecciones. Para selecciones individuales se despliega el menú mostrado en la parte izquierda de la Figura 4.14.15.

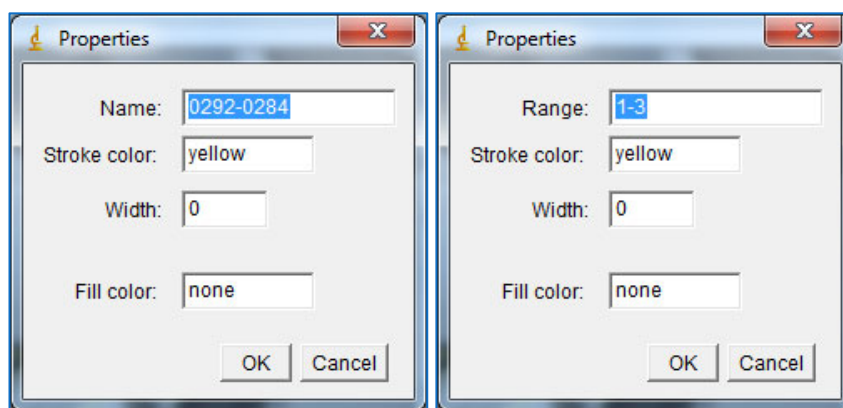


Figura 4.14.15. Izquierda: menú emergente *Properties* para selecciones individuales. Derecha: menú emergente *Properties* para grupos de selecciones

Contiene cuatro campos. El primero, *Name*, permite renombrar la selección al igual que *Rename*. En *Stroke color* es posible especificar el color de la selección: *black*, *blue*, *cyan*, *green*, *magenta*, *orange*, *red*, *white* y *yellow* para negro, azul, cian, verde, magenta, naranja, rojo, blanco y amarillo respectivamente. Cualquier otro color ha de especificarse a través de notación hexadecimal (e.g., *ffff00* para el amarillo). En *Width* se puede especificar la anchura en píxeles de la selección. Una anchura con valor 0 hace que la

selección tenga siempre un píxel de ancho sea cual sea el *zoom* empleado en la representación de la imagen (se utiliza un píxel de pantalla de ancho). El campo *Fill color* permite especificar el color con el que se rellena, a elegir entre *none* (ninguno, sin relleno) y los citados anteriormente para el campo *Stroke color*.

Cuando se seleccionan varias entradas de la lista de selecciones de forma simultánea y se pulsa el botón *Properties* se despliega el menú de la parte derecha de la Figura 4.14.15. Este menú difiere del anterior en el primer campo: en vez de ser *Name*, es *Range*, que permite especificar el rango de entradas de la lista de selecciones del gestor a las que se aplican las reconfiguraciones de propiedades.

Para terminar hay que señalar que el menú de propiedades de selecciones textuales incluye campos para determinar el tamaño de la fuente y su alineación.

- *Flatten*. Imprime de forma definitiva (invasiva) las selecciones elegidas en la lista sobre la imagen activa y presenta el resultado en una nueva ventana (se conserva la imagen original sin modificaciones en otra ventana). Una vez más, si no hay ninguna selección elegida en la lista, se imprimen todas.
- *More*. Al pulsar este botón se despliega una lista con aún más funcionalidades relacionadas con la gestión de selecciones múltiples.
 - *Open*. Permite abrir archivos con extensión *.roi* que contengan selecciones individuales o archivos *.zip* con múltiples selecciones y las incorpora al gestor de selecciones, trasladando sus nombres a la lista de selecciones.
 - *Save*. Permite guardar la selección elegida en la lista en un archivo *.roi*. Si se elige más de una selección, se guarda como un archivo comprimido *.zip*.
 - *Fill*. Rellena la selección o selecciones elegidas en la lista del gestor con el color principal (ver el apartado correspondiente a la herramienta *Color Picker*).
 - *Draw*. Imprime de forma invasiva (dibuja) las selecciones elegidas en la lista sobre la imagen activa con el color principal (ver el apartado correspondiente a la herramienta *Color Picker*). Hay que advertir que no se conserva la imagen original.
 - *AND*, *OR* y *XOR*. Genera una nueva selección actual a partir de las operaciones lógicas *AND*, *OR* o *XOR* de las selecciones elegidas (ver Figura 4.14.16). A esta nueva selección se la denomina como compuesta.

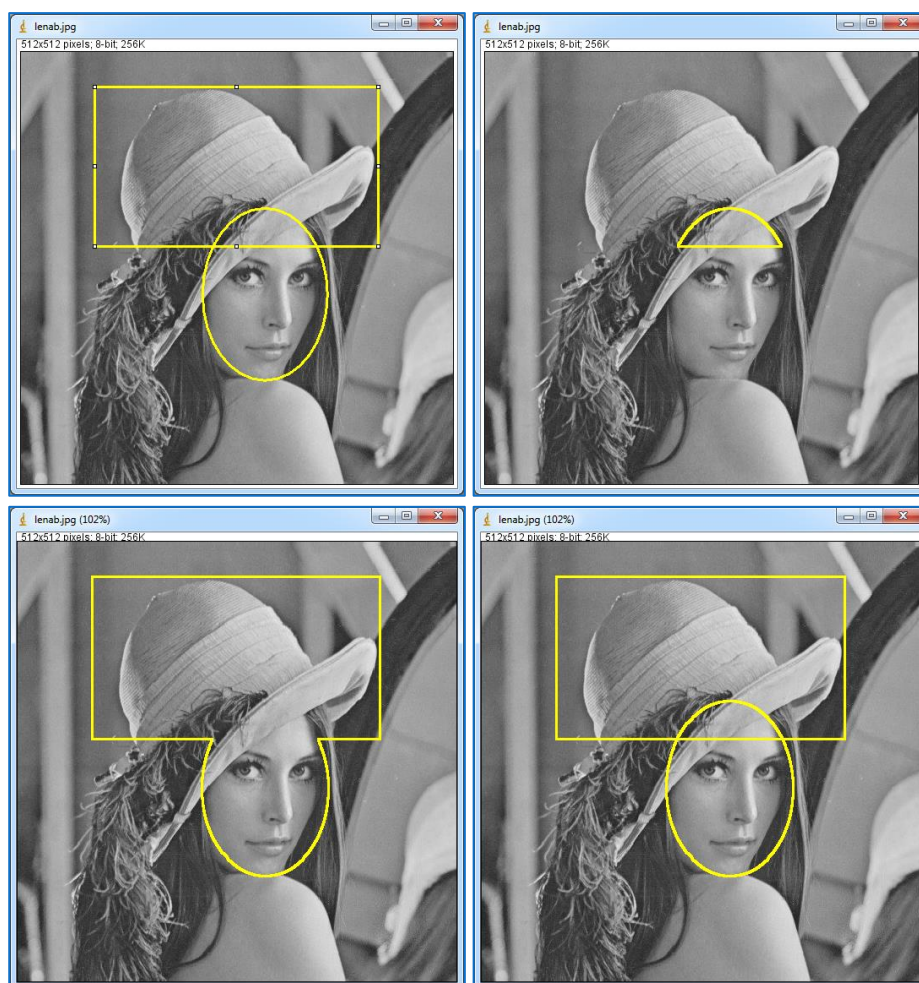


Figura 4.14.16. Arriba, izquierda: imagen con dos selecciones. Arriba, derecha: resultado de aplicar la operación AND sobre las selecciones anteriores. Abajo, izquierda: resultado de aplicar la operación OR sobre las selecciones anteriores. Abajo, derecha: resultado de aplicar la operación XOR sobre las selecciones anteriores (el segmento circular central no se incluye)

- *Split*. Divide las selecciones compuestas en sus componentes originales y añade éstas como selecciones independientes al gestor.
- *Add Particles*. Tras haber utilizado la herramienta *Analyze Particles* (con la casilla *Record Starts* marcada) permite incorporar al gestor de regiones de interés las selecciones que *Analyze Particles* ha llevado a cabo en torno las partículas.
- *Multi Measure*. Aplica la herramienta *Measure* a todas las selecciones almacenadas en el gestor. Antes de empezar la operación, se despliega una ventana que permite elegir las siguientes configuraciones: si se extiende o no la medida de todas las selecciones a todas las capas de la pila activa (evidentemente, sólo en el caso de que el elemento activo sea una pila de imágenes), con la casilla *Measure All n Slices*; si en la tabla de resultados se utiliza una fila por capa (marcando *One Row Per Slice*) o si por el contrario se utiliza una fila por medida; y por último, si se adjuntan los resultados a los anteriores conservándolos (casilla *Append results*), o se prescinde de estos últimos.

- *Multi Plot*. Aplica la herramienta *Plot Profile*, previamente comentada, a todas las selecciones elegidas en la lista del gestor (o a todas las selecciones si no hay ninguna entrada marcada en la lista) y muestra los resultados en un mismo gráfico. La herramienta *Plot Profile* sólo es aplicable a selecciones lineales, así que antes de usarla se aplica automáticamente la utilidad *Area to Line* (submenú *Selection*, menú *Edit*) sobre las selecciones que de área. Esta utilidad convierte selecciones de área a selecciones lineales desconectando el contorno de la selección en un punto. Ver Figura 4.14.17.

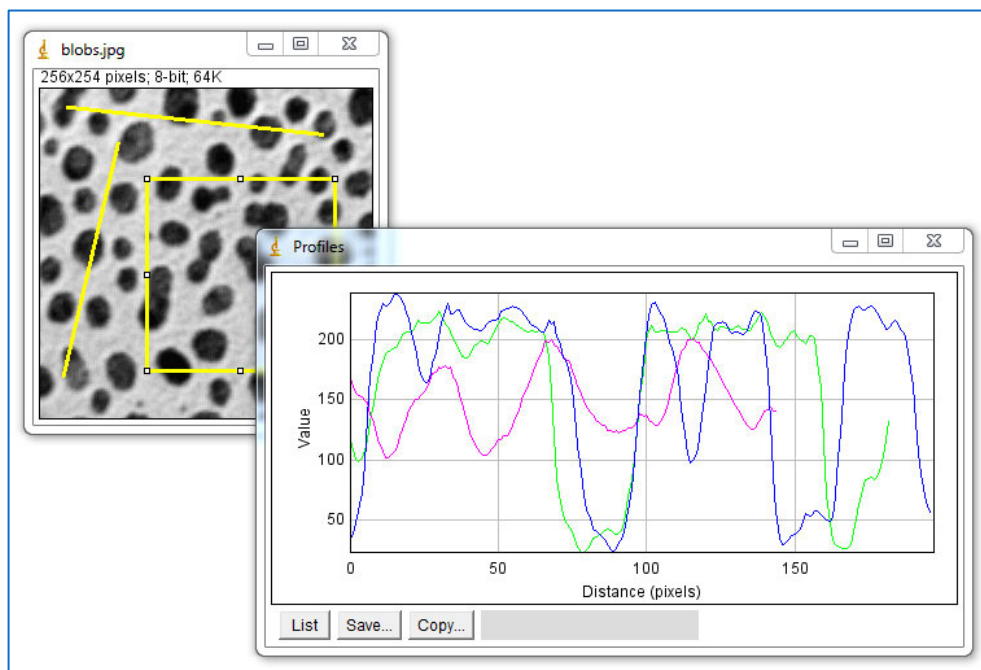


Figura 4.14.17. Selecciones múltiples (de varios tipos) sobre una imagen junto a sus perfiles representados en el mismo gráfico. Obtenidos a través de la herramienta *Multi Plot*

- *Sort*. Ordena la lista de selecciones de forma alfanumérica (0, 1, 2,...A, B, C,...a, b, c...).
- *Specify*. Despliega una interfaz con la que el usuario puede interactuar para crear selecciones rectangulares o elípticas manualmente (especificando numéricamente su altura, su anchura, las coordenadas XY de su esquina superior izquierda o de su centro,...).
- *Remove Slice Info*. Para las selecciones hechas sobre capas de pilas, elimina la primera parte de su nombre por defecto, la parte relativa al número de la capa sobre la que se seleccionó.
- *Help*. Abre una instancia del navegador por defecto del sistema y lo redirige a una página web del sitio <http://imagej.nih.gov/ij/index.html> con información detallada acerca del uso del Gestor de Regiones de Interés.
- *Labels*. Alias de la utilidad *Labels* del submenú *Overlay* (menú *Image*), comentada anteriormente.
- *List*. Despliega la ventana *ROI List*, en la cual se listan las selecciones almacenadas en el gestor junto a sus propiedades principales: índice, nombre, tipo, coordenadas XY de su centro, color,...

- *Options*. Despliega otra nueva ventana con todavía más opciones relacionadas con la gestión de selecciones múltiples.
 - *Associate "Show All" ROIs with slices*. Marcando esta casilla, la opción *Show All*, que representa todas las selecciones almacenadas en el gestor sobre la imagen actual, sólo representa las selecciones correspondientes a capas de una pila en su respectiva capa.
 - *Restore ROIs centered*. Marcando esta casilla, las selecciones recuperadas de un archivo a través de *More-Open* se recolocan de forma que aparezcan centradas en la imagen actual. De este modo se evita que las selecciones aparezcan fuera de los límites de la imagen, lo cual puede ocurrir si la imagen se ha redimensionado o si, sencillamente, se trata de otra imagen.
 - *Use ROI names as labels*. Al marcar esta casilla, los nombres de las selecciones se utilizan como etiquetas en caso de que se marque la casilla *Labels*, que hace que se impriman etiquetas junto a las selecciones del gestor que se representan sobre la imagen activa. En caso contrario, lo que se utiliza como etiquetas son los índices que tienen las selecciones en la lista del gestor.
- *Show All*. Al marcar esta casilla se representan sobre la imagen actual todas las selecciones almacenadas en el gestor.
- *Labels*. Marcando esta casilla, si se representa alguna selección sobre la imagen activa, se representa junto una etiqueta, que puede ser su índice en la lista del gestor o su nombre.

4.14.6. Scale Bar

Esta herramienta permite añadir a la imagen activa una escala (una barra de calibración espacial con su correspondiente leyenda). Despliega un menú (ver Figura 4.14.19) que permite configurar los siguientes parámetros:

- *Width in pixels/mm/ μ m/ \AA m/...* Permite especificar la longitud de la barra en píxeles o en la unidad correspondiente en caso de que la imagen se haya calibrado previamente.
- *Height in pixels*. Permite especificar el grosor en píxeles de la barra (su altura).
- *Font Size*. Permite especificar el tamaño de la letra de la leyenda que acompaña a la barra.
- *Color*. Permite elegir el color de la barra y la leyenda.
- *Background*. Permite agregar un fondo rectangular a la barra y la leyenda, y determinar además su color.
- *Location*. Permite elegir la situación de la barra y la leyenda:

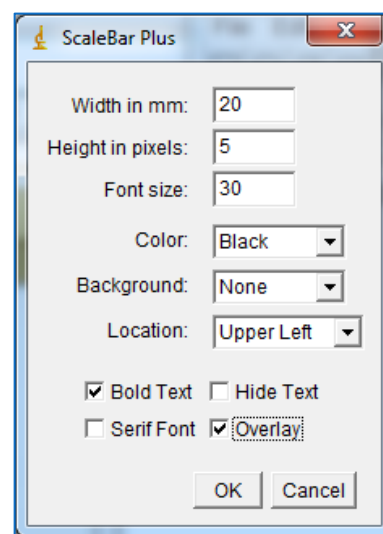


Figura 4.14.18. Menú ScaleBar Plus

Upper Right, *Lower Right*, *Lower Left*, *Upper Left* o *At Selection* (esquina superior derecha, esquina inferior derecha, esquina inferior izquierda, esquina superior izquierda o sobre la selección respectivamente).

- *Bold Text*. Marcando esta casilla, la leyenda se escribe en negrita.
- *Hide Text*. Al marcar esta casilla se oculta la leyenda.
- *Serif Font*. Añade serifas (remates) a las letras de la leyenda.
- *Overlay*. Marcando esta casilla, la barra y la leyenda se añaden al recubrimiento de la imagen, es decir, se añaden de forma no destructiva. En caso contrario se imprimen de forma invasiva (modificando los píxeles).



Figura 4.14.19. Imagen con una escala (barra y leyenda) en su esquina superior derecha

4.14.7. Calibration Bar

Calibration Bar es a *Calibrate* lo que *Scale Bar* es a *Set Scale*. Permite añadir una leyenda, que en este caso da una idea de la calibración de densidad de la imagen, y que consiste en una barra coloreada con los colores o niveles de brillo presentes la imagen y en una serie de números que se presentan equiespaciados junto a la barra y que relacionan su colores con valores concretos de la magnitud correspondientes. También despliega un menú que permite llevar a cabo las configuraciones necesarias (ver Figura 4.14.21):

- *Location*. Como en el caso de *Scale Bar*, permite elegir la situación de la leyenda.
- *Fill color*. Análogo al campo *Background* en *Scale Bar*: permite definir el color del fondo de la leyenda.
- *Label color*. Permite ajustar el color de los números escritos junto a la barra. Análogo al campo *Color* de *Scale Bar*.
- *Number of Labels*. Permite definir la cantidad de valores que se representan junto a la barra.
- *Decimal Places*. Permite definir la cantidad de decimales de los

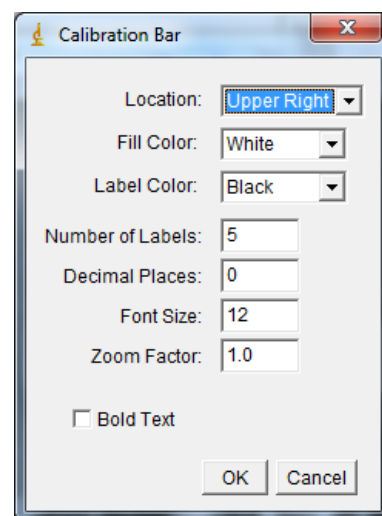


Figura 4.14.20. Menú *Calibration Bar*

números que se representan junto a la barra.

- *Font Size*. Permite definir el tamaño de los números que se representan junto a la barra.
- *Zoom Factor*. Permite agrandar (con valores superiores a 1) o reducir (con valores inferiores a 1) la barra.
- *Bold Text*. Al marcar esta casilla, los números se representan en negrita.

En la Figura 4.14.21 se muestra una imagen calibrada con una barra.

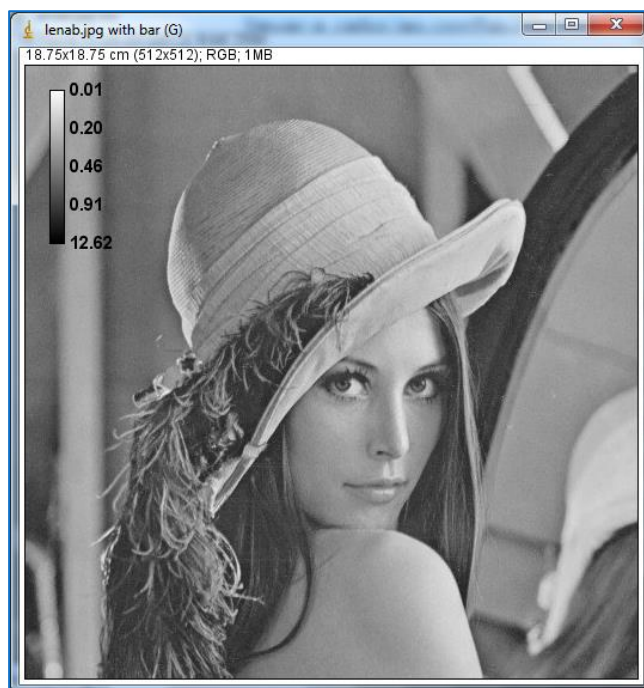


Figura 4.14.21. Imagen calibrada (en densidad óptica) con una leyenda que refleja dicha calibración

4.14.8. Synchronize Windows

Esta herramienta permite sincronizar todas las ventanas con imágenes, pilas o hiperpilas de forma que una selección hecha sobre la imagen activa se replica en las demás:

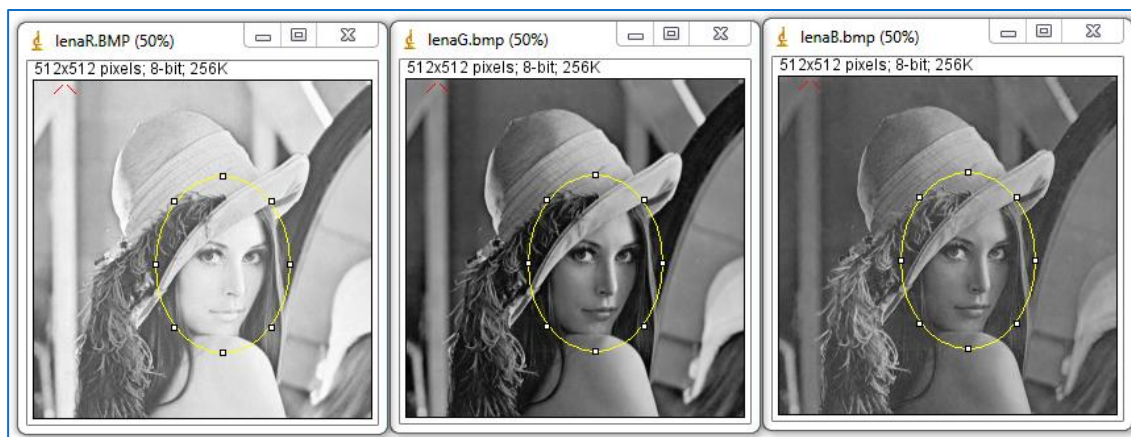


Figura 4.14.22. Selección elíptica sobre tres imágenes sincronizadas

Para configurar esta utilidad se dispone de la interfaz de la Figura 4.14.24. Se compone de los siguientes elementos:

- Lista de imágenes. Es una lista con los nombres de todas las imágenes, pilas o hiperpilas presentes en la sesión de ImageJ. Aquellas que se marcan son las que se sincronizan.

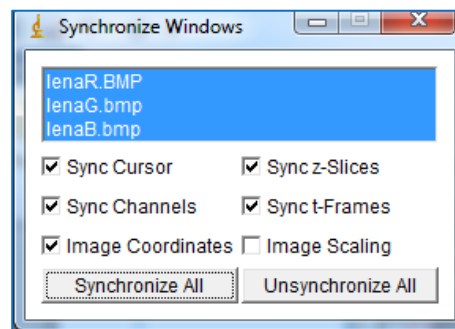


Figura 4.14.23. Menú *Synchronize Windows*

- *Sync Cursor*. Si se marca la casilla, el cursor también se replica en todas las imágenes de la sesión y todos los cursores se mueven de forma sincronizada. En caso contrario, sólo se muestra el cursor en la imagen activa.
- *Sync z-Slices*. Marcando esta casilla, se sincronizan las barras deslizantes que permiten desplazarse por las capas de las pilas sincronizadas.
- *Sync Channels*. Marcando esta casilla, se sincronizan las barras deslizantes que permiten desplazarse por las capas de las hiperpilas sincronizadas.
- *Sync t-Frames*. Marcando esta casilla, se sincronizan las barras deslizantes que permiten desplazarse por los cuadros (*frames*) de las hiperpilas sincronizadas.
- *Image Coordinates*. Para imágenes calibradas espacialmente, al marcar esta casilla se utiliza la magnitud espacial elegida en cada caso. En caso contrario, las coordenadas se dan en píxeles.
- *Image Scaling*. Al marcar esta casilla se utilizan coordenadas absolutas para sincronizar varias imágenes. En caso contrario, se utilizan coordenadas relativas a los marcos de las imágenes, lo cual puede acarrear problemas en caso de que unas imágenes estén escaladas y otras no.
- *Synchroniza All*. Al pulsar este botón se sincronizan todas las imágenes de la lista.
- *Unsyncronize All*. Al pulsar este botón se desincronizan todas las imágenes de la lista.

5. PLUGINS

En el menú *Plugins* se encuentra todo lo relacionado con la extensibilidad de ImageJ, que son, fundamentalmente, todas las herramientas relacionadas con los *macros*, los *scripts* y los *plugins*.

5.1. Submenú Macros

Un *Macro* es una “macroinstrucción” o instrucción compleja compuesta de una serie de instrucciones simples que se ejecutan secuencialmente tras una sola llamada, y que está orientada a automatizar tareas recurrentes. ImageJ provee de un lenguaje para escribir *macros* similar a Java (aunque mucho más sencillo), con funciones, operadores y estructuras de control. ImageJ también facilita la creación de *macros* a través de un grabador que registra los comandos que indica el usuario y los traduce al lenguaje *macro*. Los *macros* se guardan en forma de ficheros de texto con extensiones .txt o .ijm, y se instalan y ejecutan fácilmente con las herramientas que provee ImageJ para tal efecto, las cuales se explican a continuación. Los *macros* son la forma de extensibilidad más sencilla de ImageJ, pero presentan desventajas que cabe señalar: son más lentos e inflexibles, y sobre todo menos potentes que los *scripts* y los *plugins*.

5.1.1. Install

La herramienta *Install* permite instalar fácilmente *macros* para poder acceder a ellos directamente desde los menús. Concretamente, los *macros* recién instalados aparecen en la parte inferior del menú *Macros*. A continuación se presenta un ejemplo de instalación de un macro. En primer lugar se busca un *macro* en el apartado de *plugins* del sitio web oficial de ImageJ, <http://rsbweb.nih.gov/ij/plugins/index.html>.

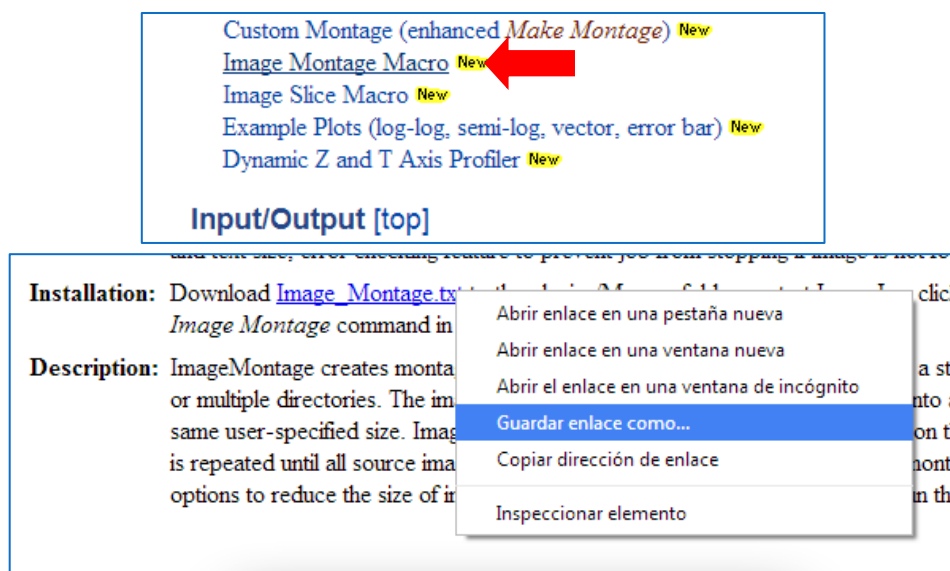


Figura 5.1.1. Arriba: macro *Image Montage* en el sitio web oficial de ImageJ. Abajo: descarga del macro *Image Montage*

Se elige uno al azar, *Image Montage*, que según su descripción sirve para crear montajes (*collages*) con imágenes. Se descarga en el escritorio como muestra la Figura 5.1.1. A continuación se usa la herramienta *Install* para instalarlo, indicado su ruta en el menú emergente. Al terminar este paso, el *macro* ha sido instalado y es accesible desde el menú *Macros*. Esto se describe gráficamente en la Figura 5.1.2:

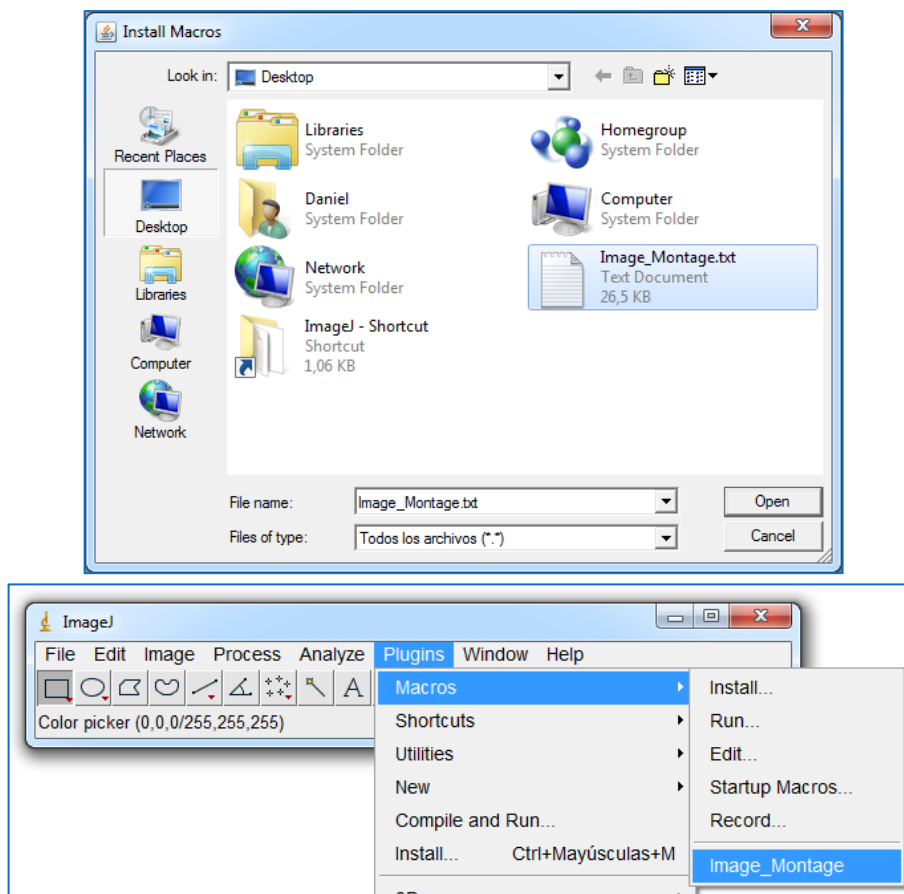


Figura 5.1.2. Arriba: instalación del *macro Image Montage*. Abajo: *macro Image Montage* accesible desde el menú *Macros*

Sin embargo, esta instalación sólo tiene efecto en la sesión actual: al cerrar ImageJ y abrirlo de nuevo, el *macro* habrá desaparecido del menú *Macros*. Hay varias formas de instalar macros de forma permanente. Una de ellas es utilizar el archivo *StartupMacros.txt* (al cual se puede acceder pulsando en *Startup Macros*, submenú *Macros*), del que se hablará más adelante. Los *macros* contenidos en este archivo se instalan de forma automática al iniciar ImageJ. *StartupMacros.txt* también contiene un *macro* especial, *AutoRun*, que se ejecuta automáticamente al cargar ImageJ (aunque por defecto aparece comentado para que no se ejecute). En vez de copiar el código del *macro Image Montage* al archivo *StartupMacros.txt*, es posible introducir el comando de instalación del *macro* dentro del *macro AutoRun* para que se instale directamente el archivo de texto .txt descargado desde el sitio web de ImageJ cada vez que se inicie ImageJ, tal y como se muestra en la Figura 5.1.3:

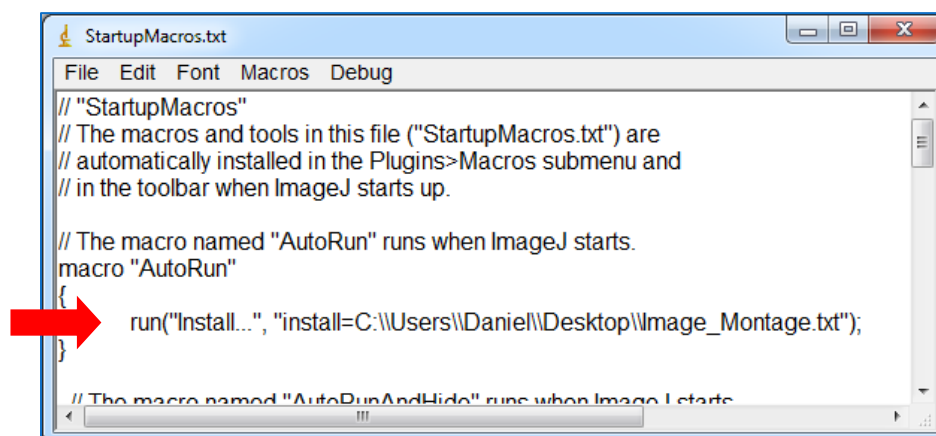


Figura 5.1.3. Archivo StartupMacros.txt con el comando necesario para que el *macro Image Montage* quede instalado de forma definitiva

5.1.2. Run

La herramienta *Run* sencillamente ejecuta un *macro* sin necesidad de instalarlo previamente. Al pulsar en *Run* se despliega un menú emergente que permite buscar en el sistema de ficheros el archivo de texto del *macro* a ejecutar. Evidentemente, si se trata de un macro muy utilizado conviene instalarlo para acceder a él más fácilmente, desde el submenú *Macros*, y no tener que buscarlo cada vez.

5.1.3. Edit

Al pulsar en *Edit* se abre un menú emergente para seleccionar el fichero de texto de un macro y después se despliega la ventana de edición, en la que es posible editar el *macro* (ver Figura 5.1.4). Para ello, obviamente, es necesario tener unas nociones básicas acerca del lenguaje *macro* de ImageJ, sobre el cual hay abundante información en la red. El sitio web oficial de ImageJ ofrece el siguiente manual:

http://imagej.nih.gov/ij/docs/macro_reference_guide.pdf.

Además, es aconsejable revisar el siguiente documento, que contiene una breve introducción al lenguaje *macro*: <http://rsb.info.nih.gov/ij/developer/macro/macros.html>

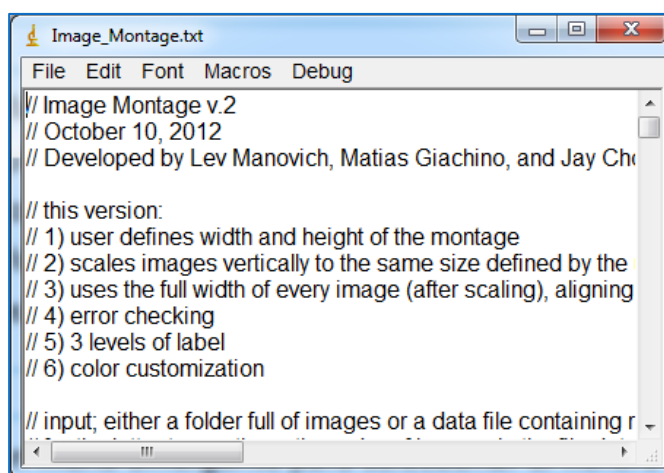


Figura 5.1.4. Ventana de edición con parte del código del *macro Image Montage*

5.1.4. Startup Macros

Startup Macros permite acceder directamente al archivo StartupMacros.txt que, como ya se ha comentado, determina los *macros* que se instalan de forma permanente en ImageJ. Esta opción evita tener que navegar por el sistema de ficheros hasta la carpeta contenedora del archivo en cuestión, que en los sistemas operativos Windows se sitúa generalmente en C:\Program Files\ImageJ\macros\StartupMacros.txt

5.1.5. Record

Record es una de las herramientas más útiles de este menú porque permite generar *macros* de forma muy sencilla, abstrayéndose de la complejidad del lenguaje *macro*: es un grabador que, desde el comienzo de la grabación, registra todos los comandos que se indican a ImageJ a través de la interfaz de usuario para traducirlos después a lenguaje *macro* y trasladarlos a un fichero de texto. A continuación, en la Figura 5.1.5, se presenta un ejemplo de su funcionamiento.

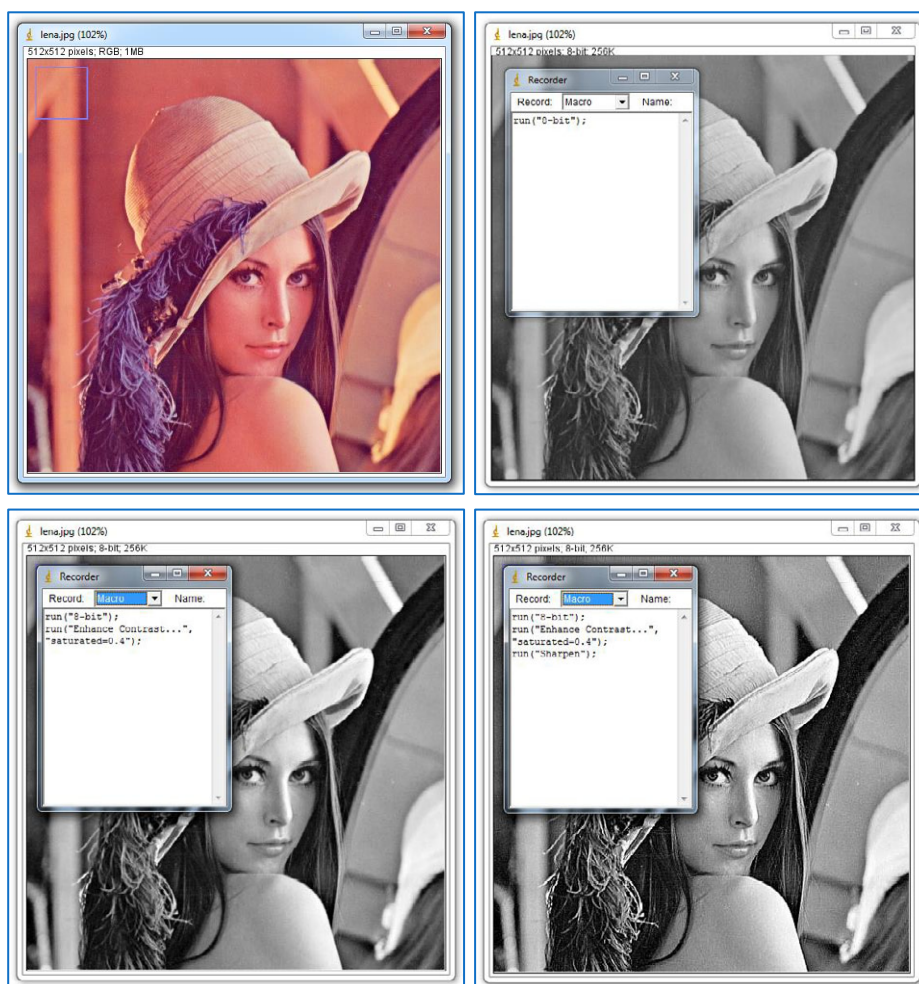


Figura 5.1.5. De arriba a abajo y de izquierda a derecha: imagen original, imagen convertida a escala de grises junto al grabador, imagen con el contraste mejorado junto al grabador, imagen con los detalles agudizados junto al grabador

Como muestra la Figura 5.1.5, se carga una imagen cualquiera y se activa el grabador. Después se convierte la imagen a escala de grises (tipo *8-bit*), se mejora su contraste con la herramienta *Enhance Contrast* y se

agudizan sus detalles con la herramienta *Sharpen*, y mientras tanto el grabador de macros va registrando toda la actividad. Una vez hecho esto, se guarda el código generado por el grabador como “Mi Macro.txt”. En la Figura 5.1.6 se muestra el archivo de texto del nuevo *macro* con los comandos generados por el grabador:

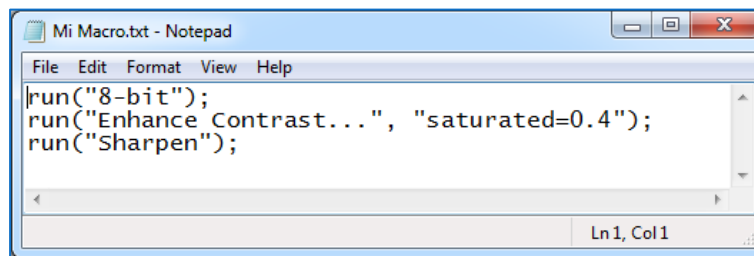


Figura 5.1.6. *Macro* generado por el grabador a partir del proceso anterior

Para verificar el funcionamiento de este nuevo *macro*, se carga otra imagen y se ejecuta sobre ella, lo cual se muestra en la Figura 5.1.7:

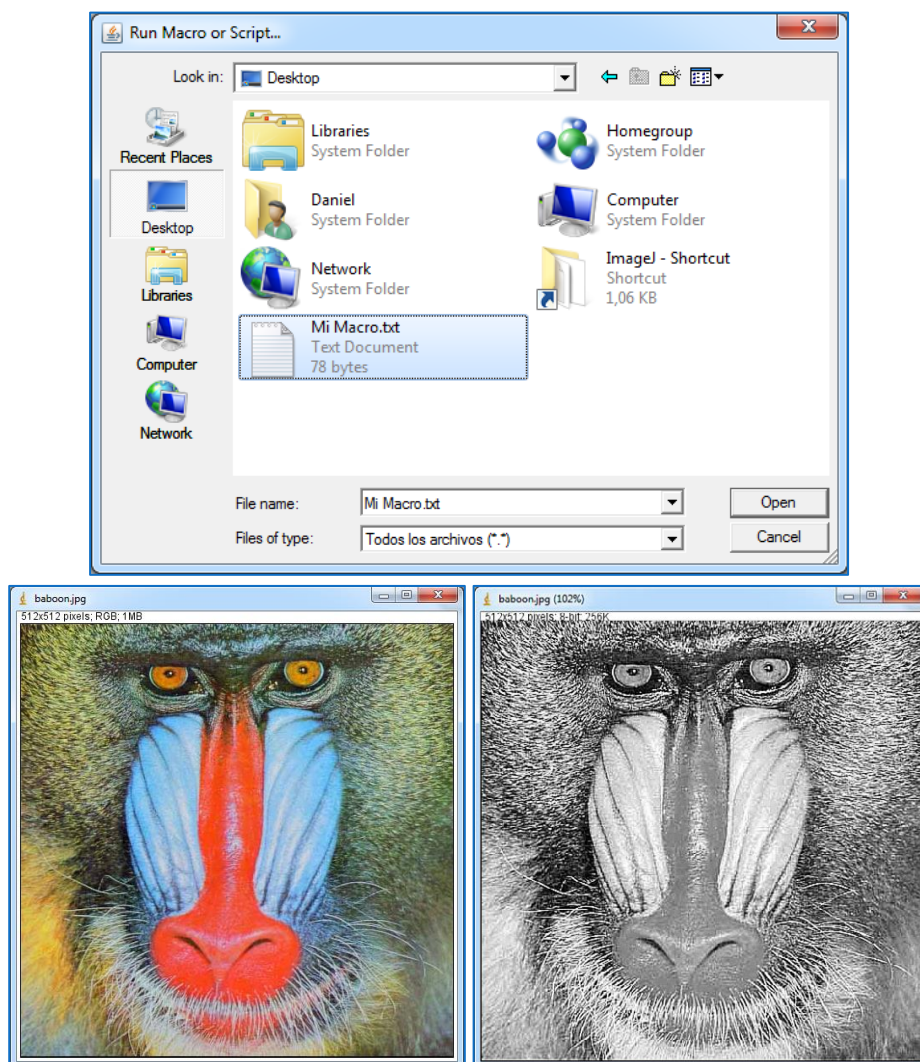


Figura 5.1.7. Arriba: carga del Nuevo *macro*. Abajo, izquierda: imagen original. Abajo, derecha: imagen resultado de aplicar el *macro* a la imagen original

En el ejemplo, el *macro* apenas consta de tres instrucciones simples, pero el grabador puede registrar comandos de forma ilimitada en cuanto a número y complejidad, lo cual resulta muy útil a la hora de automatizar tareas recurrentes.

5.2. Submenú Shortcuts

En este submenú se encuentran las herramientas necesarias para gestionar los atajos del teclado y para instalar y desinstalar *plugins*.

5.2.1. List Shortcuts

Despliega una lista (ver Figura 5.2.1) con todos los atajos de teclado configurados:

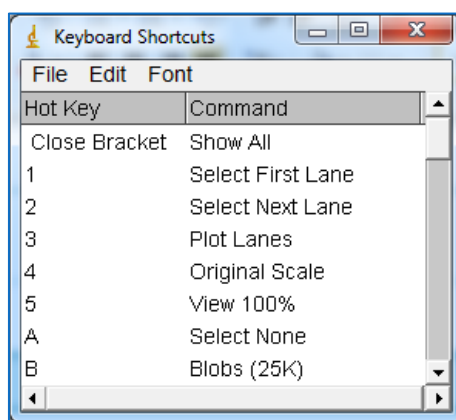


Figura 5.2.1. Lista de atajos de teclado

5.2.2. Create Shortcuts

Despliega una interfaz que permite asignar nuevos atajos de teclado a los comandos de ImageJ. Consta de una lista desplegable en la que se puede elegir el comando y de otra lista desplegable en la que se puede elegir un carácter que quede libre como atajo (letras minúsculas y mayúsculas y teclas de función desde F1 hasta F12). Los atajos de teclado añadidos aparecen en la parte inferior del submenú *Shortcuts*.

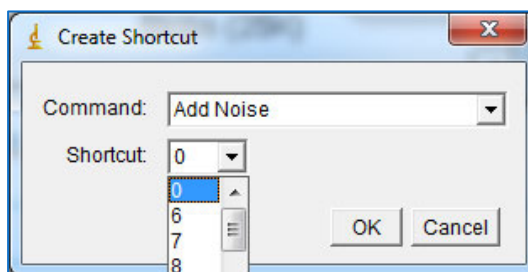


Figura 5.2.2. Menú *Create Shortcut*

5.2.3. Install Plugin

A través de *Install Plugin* es posible gestionar el acceso a los *plugins* disponibles en ImageJ. Despliega la interfaz que se muestra en la Figura 5.2.3. Esta consta de los siguientes campos:

- *Plugin*. Es una lista desplegable con todos los *plugins* disponibles en ImageJ, es decir, con todos los archivos con código byte de extensión *.class* que se encuentren en el directorio correspondiente a los *plugins* (en sistemas operativos Windows, C:\Program Files\ImageJ\plugins).
- *Menu*. Es otra lista desplegable con las distintitas posibilidades que existen (los menús disponibles) para ubicar el acceso al *plugin* elegido.
- *Command*. Este campo de texto permite especificar el comando que ha de introducirse para llamar al *plugin* (debe ser diferente de cualquier otro comando que utilice ImageJ, por lo que es recomendable verificar su disponibilidad con la utilidad *Find Commands*, comentada en las páginas siguientes).
- *Shortcut*. Campo de texto que permite especificar el atajo de teclado que se ha de usar para ejecutar el *plugin* (debe atenderse a las reglas explicadas en el apartado anterior, *Create Shortcut*).
- *Argument*. Campo opcional que permite especificar la cadena de caracteres que se pasa como argumento al método *run()* del *plugin* (ver los apartados relativos a la programación de *plugins*).

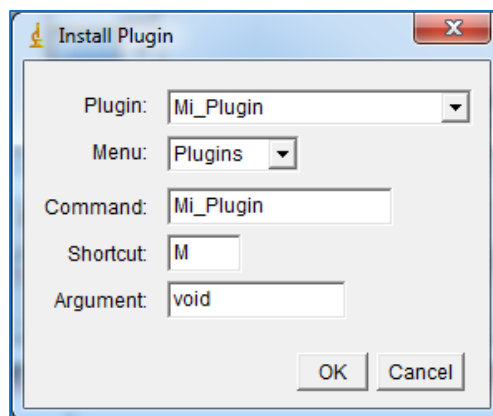


Figura 5.2.3. Menú *Install Plugin*

5.2.4. Remove

Despliega una interfaz que permite eliminar tanto los atajos de teclado añadidos (independientemente de que hayan sido añadidos a través de *Create Shortcut* o del campo correspondiente en *Install Plugin*) como los accesos a *plugins* configurados.

5.3. Submenú Utilities

En este submenú se almacena un conjunto misceláneo de utilidades relacionadas con la extensibilidad de ImageJ, su monitorización, su depuración, etc.

5.3.1. Control Panel

Control Panel despliega un panel (Figura 5.3.1) en el que se disponen todos los comandos y funciones de ImageJ organizados de forma jerárquica (tal y como se organizan los menús). Con ello se consigue agilizar el acceso a las funcionalidades que ofrece ImageJ. Para navegar a través de la jerarquía de comandos sólo hay

que expandir y plegar y los nodos. Para ejecutar un comando sólo hay que navegar hasta él y hacer doble *click* sobre su nombre. Es posible desprender una parte de la jerarquía y manejarla en una ventana aparte (una ventana “hija”). Para restaurar la jerarquía al completo desde una ventana “hija”, basta con pulsar en *Show Parent*.

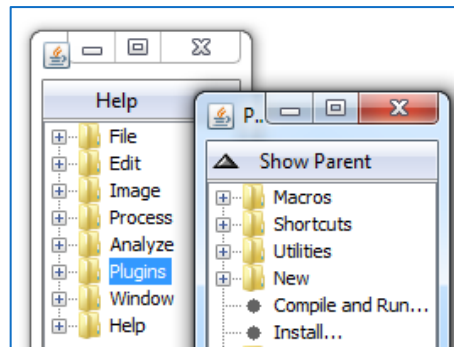


Figura 5.3.1. *Control Panel* junto a una ventana “hija”

5.3.2. Find Commands

Es un buscador de comandos que agiliza la localización de un comando en concreto. Su interfaz se muestra en la Figura 5.3.2. Al pulsar sobre *Find Commands* se despliega una interfaz con un campo de texto, *Search*, en el que se debe introducir parte del nombre (o el nombre completo) del comando de interés. Bajo este primer campo hay una tabla en la que se disponen los comandos cuyo nombre contienen la cadena de caracteres introducida. En las columnas adyacentes se proporciona información acerca de su localización, la clase Java correspondiente,... En caso de que se encuentre el comando buscado, es posible ejecutarlo desde la propia tabla haciendo doble *click* sobre su nombre. Bajo la lista hay varias casillas y botones que añaden más funcionalidades. La casilla *Close window after running command* hace que se cierre la interfaz una vez se ha ejecutado el comando. El botón *Run* ejecuta el comando seleccionado en la tabla. El botón *Source* abre el navegador por defecto y muestra en él el código fuente Java del comando seleccionado. El botón *Close* cierra la interfaz.

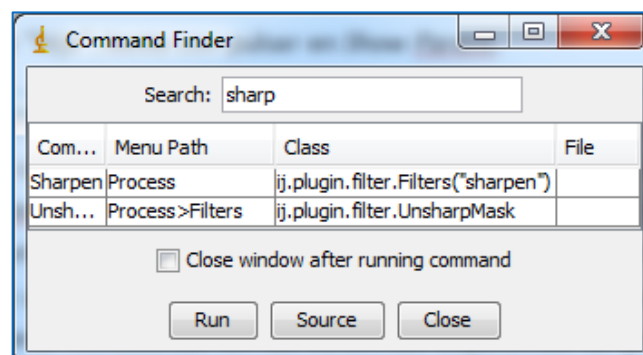


Figura 5.3.2. Interfaz del buscador de comandos

5.3.3. Search

Utilidad que permite buscar archivos correspondientes a *macros*, *scripts* y *plugins*; es decir, por un lado, archivos con extensión .txt o .ijm; por otro, archivos con extensión .js, .py, .rb, .clj o .bsh; y por último, archivos con extensión .java. Para ello, provee de la sencilla interfaz que se muestra en la Figura 5.3.3. En el campo de texto superior se ha de introducir el término de búsqueda. El campo *Search* permite elegir una de entre varias opciones de búsqueda:

- *Macros*. Busca archivos con extensión .txt o .ijm. en los subdirectorios correspondientes a *macros*, *scripts* y *plugins* (C:\Program Files\Image\macros\, C:\Program Files\Image\scripts\ y C:\Program Files\Image\plugins\ respectivamente)
- *Scripts*. Busca archivos con extensión .js, .py, .rb, .clj o .bsh. en los subdirectorios ya mencionados.
- *Java*. Busca archivos fuente de Java (.java) en los subdirectorios ya mencionados.
- *ImageJ folder*. Busca a partir del directorio de ImageJ (C:\Program Files\ImageJ\)) archivos con cualquier extensión cuyo nombre contenga el término de búsqueda.
- *Choose*. Busca a partir del directorio indicado archivos con cualquier extensión cuyo nombre contenga el término de búsqueda.

Además hay dos casillas, *Search contents* e *Ignore case*. Al marcar la primera, las búsquedas se extienden a aquellos ficheros de texto entre cuyo contenido aparezca el término de búsqueda. Al marcar la segunda, la búsqueda se hace insensible a mayúsculas y minúsculas.

Por último, hay dos botones *OK* y *Cancel*. Al pulsar el primero se inicia el proceso de búsqueda, y en caso de que haya resultados se muestran en la ventana *Log*, que se despliega a tal efecto. El botón *Cancel* cierra la interfaz.

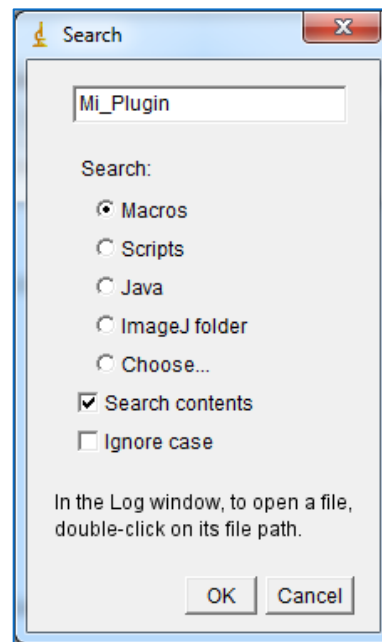


Figura 5.3.3. Menú Search

5.3.4. Monitor Events

Esta herramienta permite monitorear una gran parte de las interacciones que el usuario puede llevar a cabo con ImageJ, las cuales registra en tiempo real en la ventana *Log*, como muestra la Figura 5.3.4. La clase Java correspondiente implementa las interfaces *IJEventListener*, *CommandListener* e *Image Listener*, lo cual asegura que es capaz de monitorear cambios en los colores principal y de fondo, cambios de herramienta, cierres de

ventanas de registro, ejecuciones de comando, cargas de imágenes, cierres de imágenes y actualizaciones de imágenes.

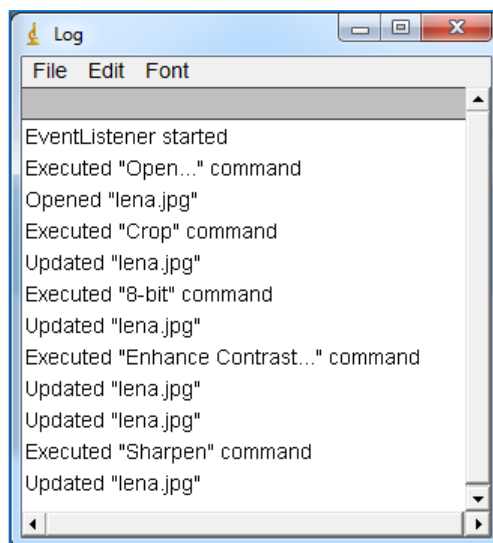


Figura 5.3.4. Registro de la monitorización de una sesión de ImageJ resultado de emplear la utilidad *Monitor Events*

5.3.5. Monitor Memory

Esta herramienta permite monitorear en tiempo real el uso de la memoria que hace ImageJ. Consiste en un pequeño gráfico con el consumo de memoria en el eje de ordenadas y el tiempo en el eje de abscisas (el extremo derecho corresponde al instante inmediatamente anterior a la inspección). Además de la correspondiente curva, en la esquina superior izquierda se muestran los consumos de memoria absoluto y porcentual. En la esquina superior derecha se muestra el número de hilos activos (se recuerda que ImageJ se trata de un programa multiproceso). Además, al hacer *click* en el monitor se reclama la memoria que no se está usando ejecutando el recolector de basura Java (este *daemon* compacta la memoria a base de colocar los objetos en uso al principio y liberar los huecos intermedios correspondientes a objetos extintos).

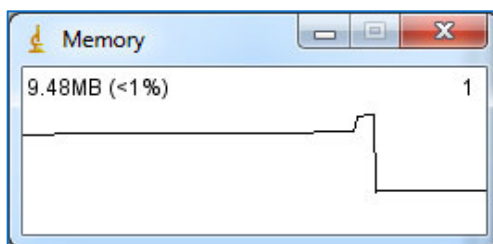


Figura 5.3.5. Monitor del uso de memoria de ImageJ

5.3.6. Capture Screen

Hace una captura de pantalla y muestra el resultado en una nueva ventana como una imagen de tipo *RGB*.

5.3.7. Capture Image

Hace una captura de la imagen activa y muestra el resultado en una nueva ventana como una imagen de tipo *RGB*.

5.3.8. ImageJ Properties

Abre una nueva ventana en la que se listan algunas propiedades de la instancia concreta de ImageJ: versiones de Java, sistema operativo anfitrión, usuario, versión de ImageJ,...

5.3.9. Threads

Abre una ventana en la que se listan los hilos en funcionamiento junto a sus prioridades.

5.3.10. Benchmark

Esta utilidad permite hacer una estimación del rendimiento de ImageJ. Para ello ejecuta 62 operaciones de procesamiento sobre la imagen activa y muestra el tiempo que ha necesitado para ello en la barra de estado:

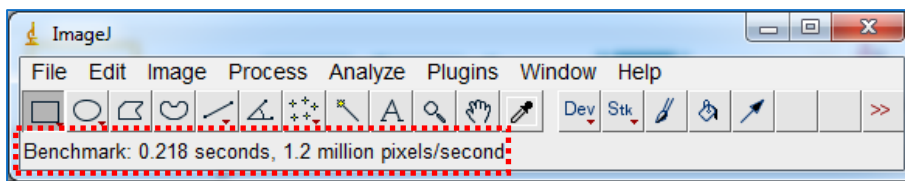


Figura 5.3.6. Destacado en rojo, resultado del *benchmark*

5.3.11. Reset

Esta utilidad consiste en una pequeña ventana que permite llevar a cabo las siguientes operaciones:

- *Reset Locked Image.* Permite desbloquear una imagen. Las imágenes pueden ser bloqueadas, por ejemplo, por *plugins* de tipo filtro, que impiden el acceso a las imágenes que procesan mientras se lleva a cabo el procesamiento.
- *Reset Clipboard.* Vacía el portapapeles de ImageJ.
- *Reset Undo Buffer.* El *buffer Undo* (deshacer) almacena la última operación que se llevó a cabo sobre la imagen para poder revertirla. *Reset Undo Buffer* vacía este *buffer*.

5.4. Submenú New

Las utilidades agrupadas en el submenú *New* permiten crear nuevos *macros*, *scripts* y *plugins* en el editor de ImageJ. Además, hay ciertas herramientas que proveen del código de prototipos de *macros* y *plugins* específicos.

5.4.1. Macro

Al pulsar en *macro* se abre una ventana de edición optimizada para *macros* (se guarda con extensión *.txt*, tiene un menú con utilidades específicas de los macros,...). Como ya se ha comentado, para escribir un *macro* es conveniente conocer la sintaxis del lenguaje *macro* de ImageJ y sus comandos. En el siguiente ejemplo se muestra un sencillo *macro* que es el resultado de mejorar el *macro* grabado en el apartado *Recorder* (submenú *Macro*):

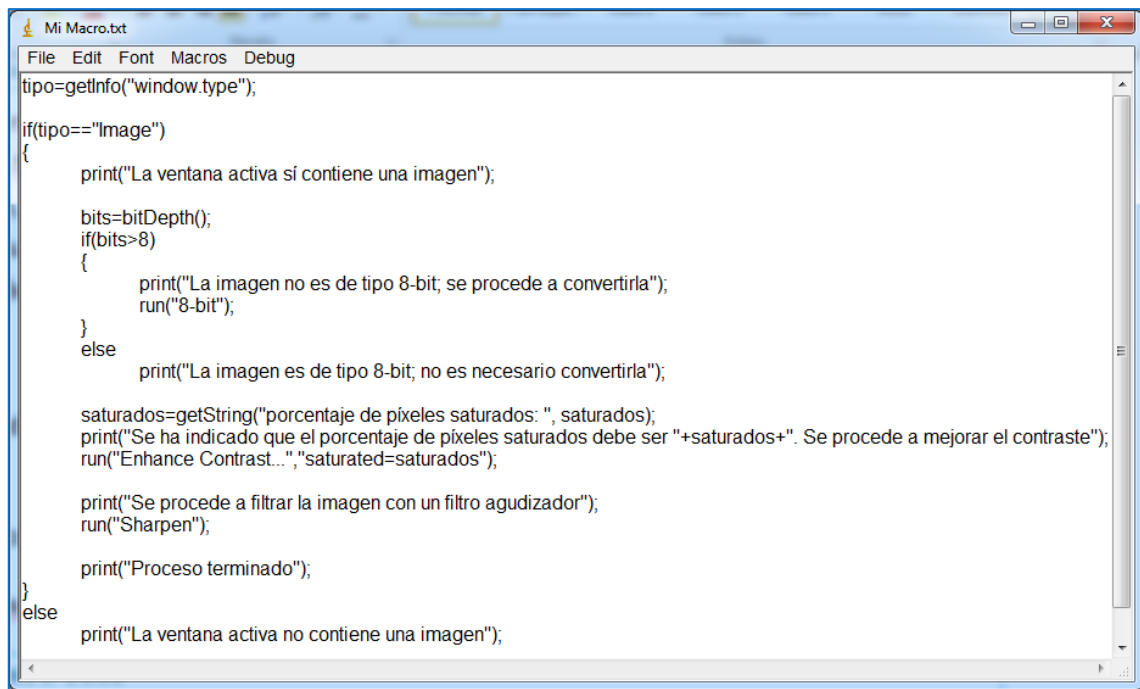


Figura 5.4.1. Macro sencillo programado en el editor de ImageJ

La primera sentencia del *macro* de la Figura 5.4.1 guarda en la variable “tipo” el tipo de la ventana activa utilizando la función *getInfo()* con el argumento *window.type*. Como se puede observar, en el lenguaje *macro* las variables no tienen que ser declaradas y no tienen un tipo explícito. Se inicializan de forma automática cuando aparecen en una sentencia de asignación. A continuación se introduce una sentencia condicional con los términos *if* y *else* y los operadores correspondientes: si el tipo de la ventana es imagen (*Image*), se muestra un mensaje de depuración positivo en la ventana *Log* con la función *print()* y se sigue adelante con el resto de instrucciones del *macro*; en caso contrario (si es una tabla de resultados, un histograma, un texto,...), se salta al *else*, se muestra un mensaje de error y se termina la ejecución. Si, efectivamente, la ventana contiene una imagen, se comprueba que ésta sea de tipo *8-bit* con otra sentencia condicional, y en caso contrario se realiza la conversión con la función *run()*, que ejecuta el comando que se le pasa como argumento (en este caso, “8-bit”, que hace la conversión a tipo *8-bit*). De nuevo se vuelven a registrar mensajes de depuración en la ventana *Log*. A continuación se pide al usuario que especifique el porcentaje de píxeles que se han de saturar en la mejora de contraste que se ejecuta después. Para ello se utiliza la función *getString()*, que recibe la cadena de caracteres introducida por el usuario en una ventana emergente y la guarda en una nueva variable (en este caso, *saturados*). Después se lleva a cabo la mejora de contraste ejecutando el comando *Enhance Contrast* con la función *run(command, arguments)*. Este comando necesita recibir como parámetro el porcentaje de píxeles saturados, así que se introduce como segundo argumento de la función *run(command, arguments)*, a través de la variable *saturados*. Una vez más, se vuelve a registrar toda la operación en la ventana *Log*. Para terminar se ejecuta un último comando, que ésta vez no necesita parámetros, *Sharpen*, para agudizar

los detalles de la imagen. En la Figura 5.4.2 se muestra el resultado de ejecutar el *macro* cuando en la ventana activa se encuentra la imagen tipo *RGB Color* lena.jpg:

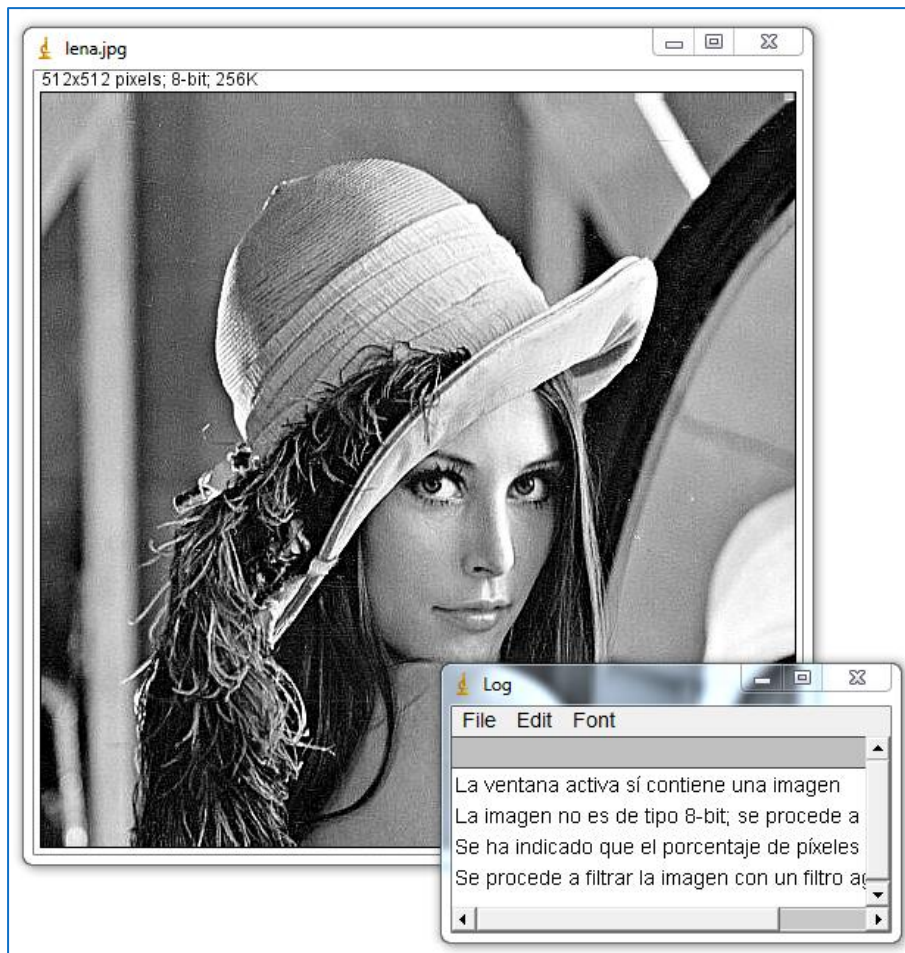


Figura 5.4.2. Resultado de aplicar el *macro* anterior a la imagen lena.jpg (de tipo *RGB*)

5.4.2. Macro Tool

Al pulsar en *Macro Tool* también se despliega la ventana de edición optimizada para *macros*, pero a diferencia de lo que ocurre al pulsar en *Macro*, en este caso la ventana de edición contiene el código del prototipo de un *macro* de tipo *Macro Tool* (herramienta *macro*). Este tipo de *macros* se representan por un icono en la barra de herramientas de *ImageJ*. El código de ejemplo que se muestra en la Figura 5.4.3 corresponde a una herramienta para dibujar selecciones circulares perfectas del radio especificado en el punto que se determine a través del cursor.

Este *macro* es en realidad un *macro set*, que es un archivo de texto en el que se definen más de un *macro* que se relacionan a través de variables externas o globales, que son un tipo de variable especial en este lenguaje *macro*. La primera línea después de los comentarios (todas aquellas líneas que comienzan por “//”) sirve para definir e inicializar esta variable global, para lo cual se emplea el término *var*. En este caso, la variable global se llama *radius* y almacena el valor del radio de los círculos que se dibujan, que por defecto vale 20 píxeles (o la unidad

correspondiente si se emplea sobre una imagen calibrada). La siguiente sentencia sirve para definir el comienzo del primer *macro* de este *macro set*. Para ello se emplea el término *macro* seguido del nombre del *macro* entrecomillado, que en este caso es *Circle Tool*.

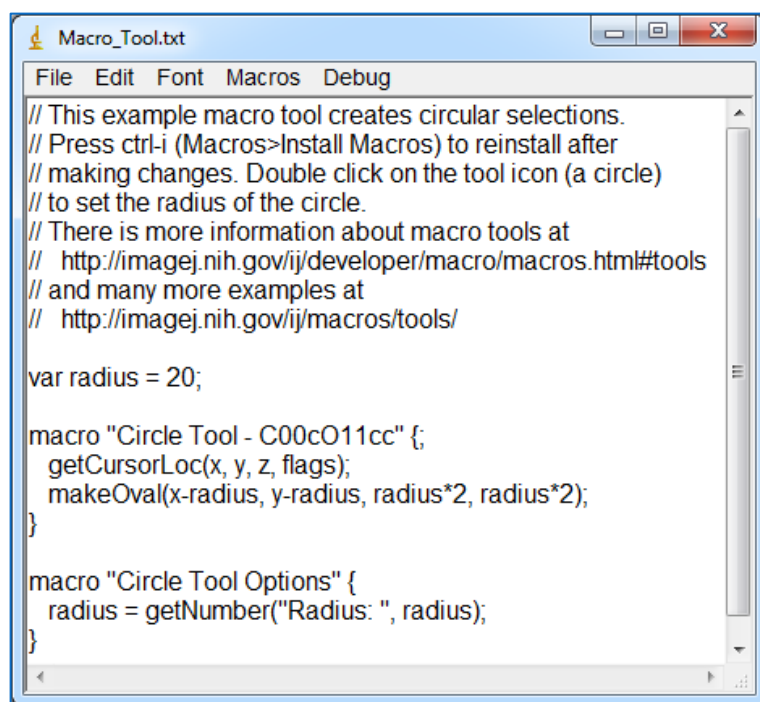


Figura 5.4.3. Código del prototipo de herramienta *macro* que se presenta en la ventana de edición al hacer *click* en *Macro Tool*

Como se trata de una herramienta *macro* y no de un *macro* clásico, al nombre le siguen un guion y un código que sirve para definir el icono que aparecerá en la barra de herramientas de ImageJ representando al *macro*. La tabla de la Figura 5.4.4 describe dicho código:

Command	Description
Bxy	set base location (default is 0,0)
Crgb	set color
Dxy	draw dot
Fxywh	draw filled rectangle
Gxyxy...xy00	draw polygon (requires 1.48j)
Hxyxy...xy00	draw filled polygon (requires 1.48k)
Lxyxy	draw line
Oxywh	draw oval
Pxyxy...xy0	draw polyline
Rxywh	draw rectangle
Txysc	draw character
Vxywh	draw filled oval (requires 1.48j)

Figura 5.4.4. Descripción del código que acompaña al nombre de los *macro tools* y que define su icono

La siguiente sentencia hace uso de la función *getCursorLoc(x,y,z,modifiers)* para registrar la posición y los *flags* del cursor cuando se hace *click* y guardarlo todo en las variables *x,y,z* y *flags*. Después se utiliza la función *makeOval(x,y,width,height)*, que sirve para crear selecciones elípticas en general, para generar la selección circular. Los argumentos *x* e *y* de *makeOval(x,y,width,height)* se refieren a la esquina superior izquierda de la elipse, y de ahí que se indique que la esquina se tiene que situar en *x-radius* e *y-radius* (para que la selección circular quede centrada en el punto de la imagen sobre el que se hizo *click* al principio). Además, *width* y *height* son la altura y la anchura de la elipse, que en este caso es una circunferencia, y por tanto valen lo mismo, que es el doble del radio: *radius*2*. Así termina el primer *macro* de este *macro set*. Cualquier herramienta, y por tanto también un *macro tool*, puede desplegar un menú de opciones cuando se hace doble *click* o *click* derecho sobre su icono. Para configurar este menú en el caso de un *macro tool* hay que añadir un segundo *macro* a su correspondiente *macro set* con el mismo nombre que el *macro* principal seguido del término *Options*, que es lo que ocurre en el *macro set* de la Figura 5.4.3. En este caso, el menú de opciones del *macro tool* Circle Tool sirve para determinar el radio de las selecciones circulares, y por eso se usa la función *getNumber("prompt", defaultValue)*, que muestra un cuadro de diálogo con el texto introducido como primer argumento entrecomillado y con un campo de texto en el que el usuario puede introducir un número (por defecto aparece el número que se pase como segundo argumento, o *defaultValue*). El valor introducido se almacena en la variable global *radius* para que pueda ser utilizado por el primer *macro*.

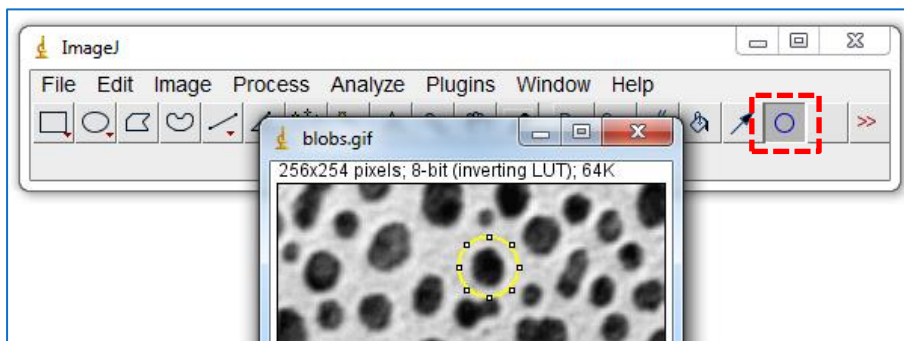


Figura 5.4.5. Selección circular generada con el *macro tool* anterior, cuyo icono se destaca en rojo

5.4.3. JavaScript

JavaScript es un lenguaje de programación. No es un lenguaje compilado, sino interpretado; es decir, se ejecuta a través de un intérprete, un programa que realiza la traducción del código a medida que es necesaria, al contrario que los compiladores (habitualmente, la traducción se lleva a cabo instrucción a instrucción). “Es un lenguaje orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico” [9]. Su sintaxis es similar a la de C y también en parte a la de Java, aunque a pesar de esto y de su nombre, JavaScript no está directamente relacionado con Java. Es utilizado principalmente en la creación de páginas web dinámicas.

ImageJ permite utilizar *scripts* (programas JavaScript) para permitir el acceso a sus funcionalidades de una forma alternativa a los *macros* y los *plugins*. Para poder interpretar los *scripts*, ImageJ hace uso del intérprete de código abierto Mozilla Rhino, el cual está programado en Java. Los *scripts* van un paso más allá que los *macros*, y permiten interactuar con la API de ImageJ y utilizar un lenguaje más rico, potente y flexible, con objetos, operadores más avanzados, estructuras más sofisticadas, etc. Sin embargo, la necesidad de traducir el *script* a medida que se ejecuta hace que su rendimiento y su velocidad sean peores que los de los *plugins*, y además tampoco cuentan con otras ventajas con las que sí cuentan los *macros* (no hay un equivalente *script* del *macro set*, no son compatibles con el procesamiento por lotes, no pueden generar herramientas que se integren en la barra de herramientas, su programación es más compleja,...).

Al igual que en el caso de los *macros* y los *plugins*, hay abundante información sobre ellos disponible y también un repositorio de *scripts*, el cual se puede encontrar en la siguiente dirección: <http://imagej.nih.gov/ij/macros/js/>. Su instalación es tan sencilla como la de los macros (de hecho se utilizan las mismas herramientas), y en cualquier caso se pueden ejecutar directamente a partir del fichero correspondiente. A continuación (Figura 5.4.6) se muestra un ejemplo de la instalación de *scripts* en ImageJ:

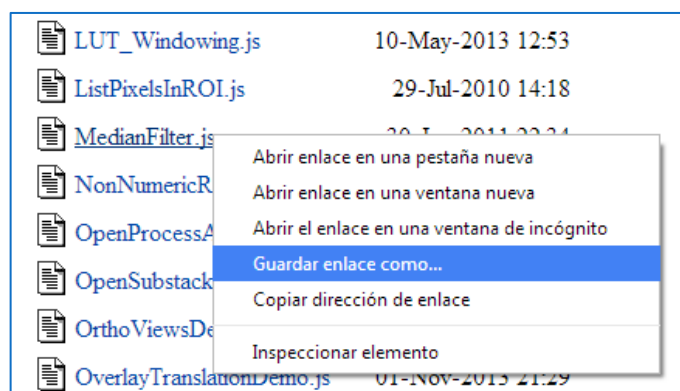


Figura 5.4.6. Repositorio de *scripts* del sitio oficial de ImageJ

En primer lugar se accede a la mencionada página web y se elige un *script*. En este caso, se elige el *script* *MedianFilter.js*, que hace uso de las funcionalidades que ofrece ImageJ para aplicar un filtro de mediana a la imagen activa. Para instalarlo basta con trasladar su fichero al directorio de *plugins* de ImageJ (Figura 5.4.7) y reiniciar el programa.

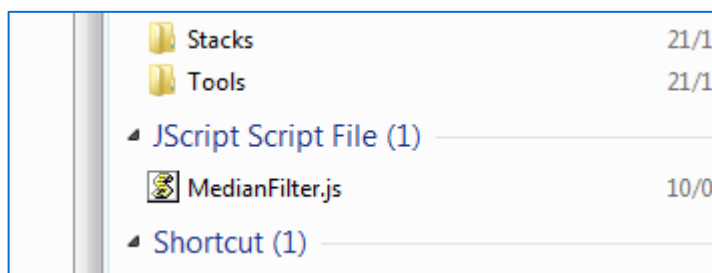


Figura 5.4.7. *Script* en el directorio de *plugins* de ImageJ

Una vez se ha reiniciado ImageJ, el *script* debería ser accesible desde el menú *Plugins* (Figura 5.4.8):

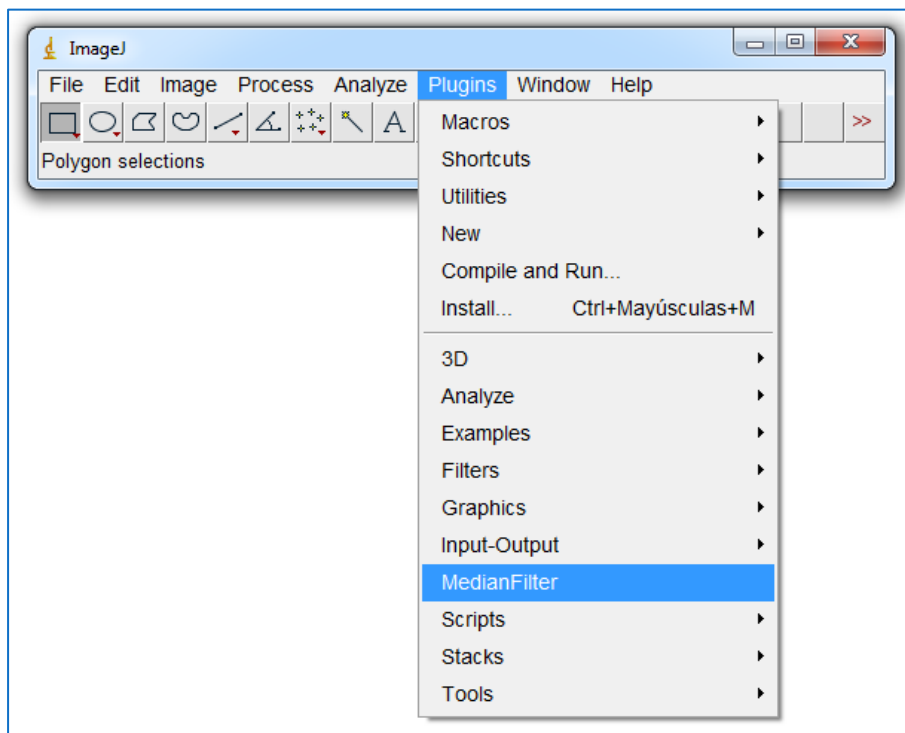


Figura 5.4.8. Acceso al *script* instalado en el menú *Plugins*

Como ya se ha comentado, también existe la opción de ejecutar el *script* directamente desde su fichero, sin necesidad de instalarlo en ImageJ. Para ello se utiliza la misma herramienta que en el caso de los *macros*: *Run*, en el submenú *Macros*.

Una vez introducidos los *scripts*, se pasa a comentar la herramienta que da nombre a este apartado. *JavaScript*, en el menú *New*, abre la ventana de edición preparada para la programación de *scripts*. Sin embargo, a diferencia de otras herramientas como *Plugin Filter* o *Plugin Frame*, no se proporciona el código prototipo, sino que la ventana aparece en blanco. Por eso, en el ejemplo (Figura 5.4.9), se ha reprogramado en la ventana de edición el *script* *MedianFilter.js*, para que además aparezca un pequeño ejemplo de código JavaScript y su aplicación en ImageJ:

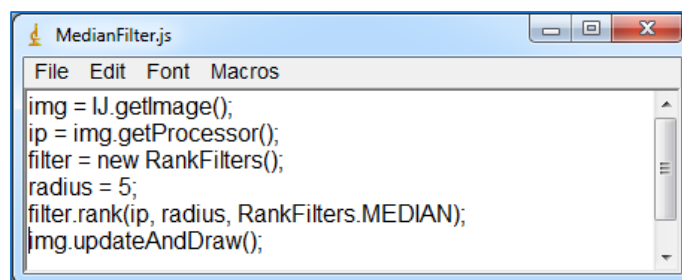


Figura 5.4.9. Ventana de edición con el código del *script* *MedianFilter.js*

En el *script* de la figura anterior, la primera sentencia hace uso de la API de ImageJ (concretamente, se hace uso del método estático *IJ.getImage()* de la clase *IJ*) para almacenar la imagen actual en el objeto *ImagePlus img*. A continuación, se obtiene la información de sus píxeles y se almacena en el objeto *ImageProcessor ip* (todas estas clases de la API de ImageJ se comentan más detalladamente en los siguientes apartados). Después se construye un filtro, *filter*, con *RankFilters*, que a priori es capaz de hacer filtrados de media, de mínimo, de máximo, de mediana,... Con la siguiente sentencia se especifica el radio del filtro, y a continuación tiene lugar el filtrado de *mediana* con el radio especificado de los píxeles cuya información se almacena en *ip*. Para terminar, se actualiza la información de los píxeles y se representa de nuevo la imagen. En la Figura 5.4.10 se muestra el resultado de aplicar este *script*:

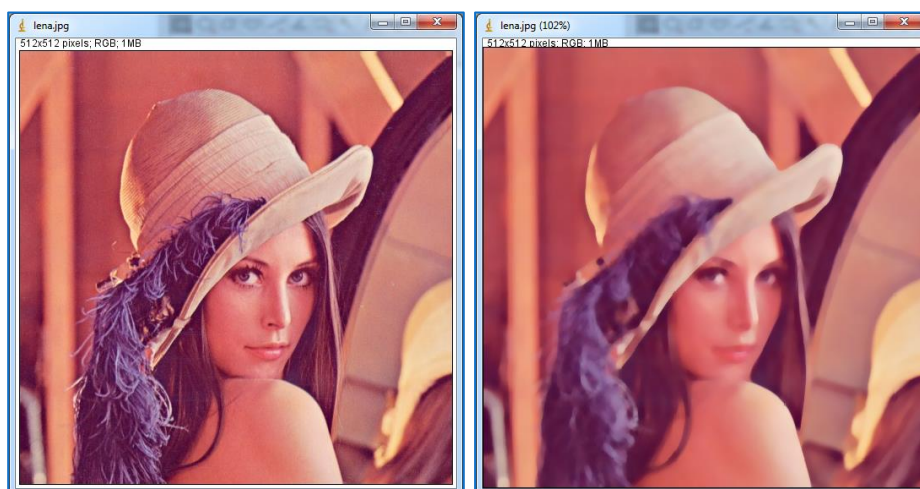


Figura 5.4.10. Izquierda: imagen original. Derecha: imagen original tras aplicarle el *script MedianFilter.js*

5.4.4. Plugin

Un *plugin* es una aplicación que complementa a la aplicación anfitriona (en este caso, *ImageJ*), aportándole una funcionalidad nueva. El *plugin* es ejecutado por la aplicación principal y ambas se relacionan a través de una API (*Application Programming Interface*, Interfaz de programación de aplicaciones), que es una colección de métodos que ofrece una biblioteca para que sean utilizados por otra aplicación [10]. ImageJ tiene su propia API, la cual se ofrece libremente para que los desarrolladores interesados puedan desarrollar sus propios *plugins* y éstos se integren fácilmente en ImageJ. La API de ImageJ se puede encontrar en su sitio web oficial: <http://rsbweb.nih.gov/ij/developer/api/index.html>. Los *plugins* de ImageJ se implementan como clases Java, las cuales resultan de compilar archivos fuente .java. La forma en que se implementan estos *plugins* hace posible aprovechar todos los recursos del lenguaje Java, de la API de ImageJ y de cualquier otra API externa que se desee utilizar, así que esta forma de extensibilidad de ImageJ, aunque más compleja, es mucho más potente, flexible y rápida que los *macros* y los *scripts*. De hecho, muchas funcionalidades *built-in* de ImageJ son en realidad *plugins*.

ImageJ ofrece en su sitio web oficial una gran cantidad de *plugins*, cuyo número además aumenta a diario debido a la implicación de la comunidad de desarrolladores de ImageJ. Puesto que ImageJ nació enfocado al tratamiento digital de imagen aplicado a las ciencias de la salud, una gran parte de los *plugins* son aplicables a prácticas médicas muy específicas, pero hay también una gran cantidad de *plugins* que encuentran aplicación en áreas del tratamiento digital de la imagen totalmente diferentes. Hay también varios paquetes de *software* basados en ImageJ que incorporan por defecto diferentes combinaciones de los *plugins* que hay disponibles en función de su orientación. El más exitoso de estos paquetes es Fiji (*Fiji Is Just ImageJ*) que en su sitio web se autodefine como sigue:

*“Fiji compares to ImageJ as Ubuntu compares to Linux”
(Fiji es a ImageJ lo que Ubuntu es a Linux)*

En este mismo sitio web se puede conseguir de forma gratuita: <http://fiji.sc/Fiji>

La instalación de *plugins* en ImageJ es muy sencilla; a continuación se muestra un ejemplo. Se empieza por buscar un *plugin* en el apartado de *plugins* del sitio web de ImageJ (<http://rsbweb.nih.gov/ij/plugins/index.html>). Se elige uno al azar, *AVI Reader* (ver Figura 5.4.11). Como se explica en su descripción, este *plugin* carga películas AVI en forma de pilas de imágenes.

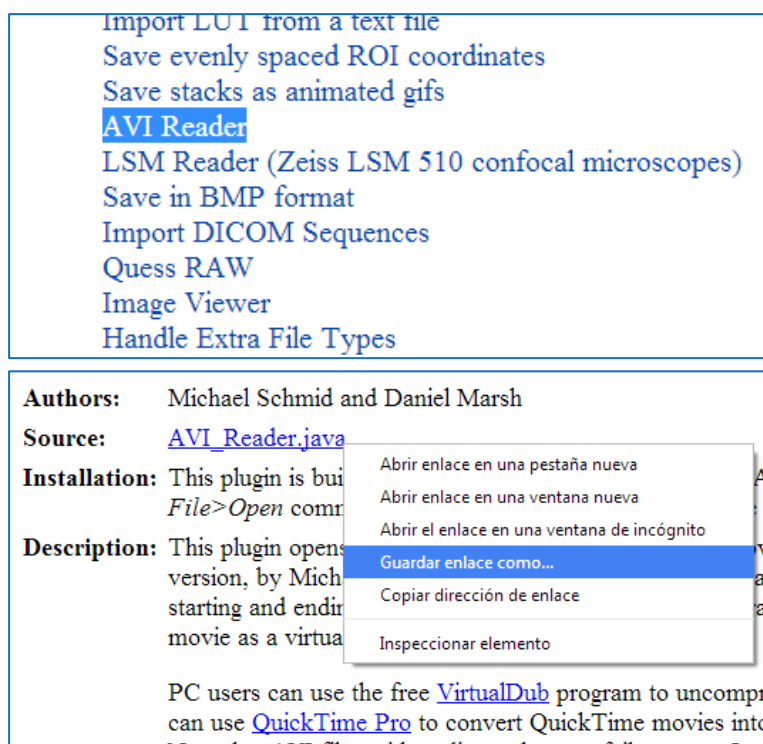


Figura 5.4.11. Arriba: *plugin AVI Reader* en el sitio web oficial de ImageJ. Abajo: descarga del *plugin AVI Reader*

Se selecciona y se descarga. En este caso no se proporciona el fichero .class, sino el código fuente en un fichero .java. Para conseguir el fichero .class hay que compilar el código fuente, lo cual se puede hacer desde ImageJ con la herramienta *Compile and Run* del menú *Plugins*, como muestra la Figura 5.4.12:

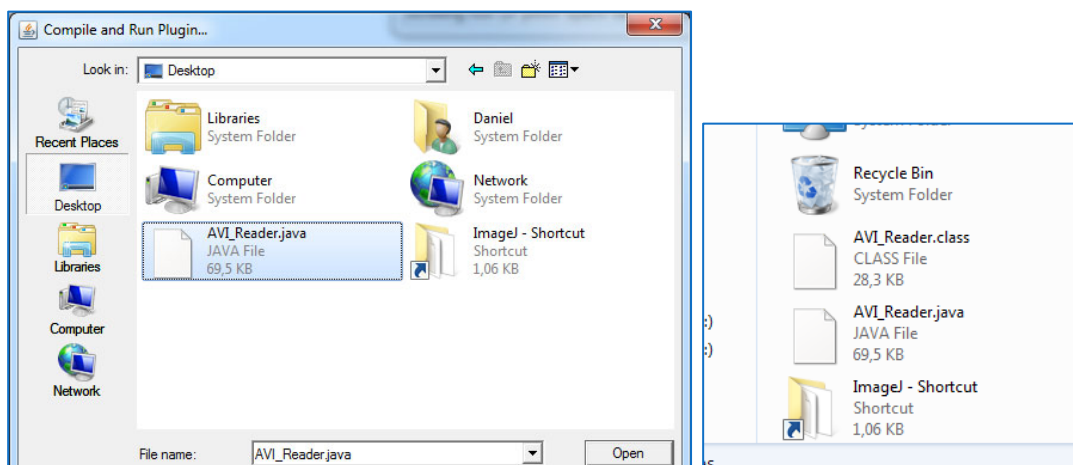


Figura 5.4.12. Izquierda: compilación del código fuente del *plugin AVI Reader*. Derecha: fichero .class del *plugin AVI Reader* junto a su código fuente

Como resultado de emplear esta herramienta, se genera el fichero .class en la misma ubicación en la que se encontraba el código fuente (en este caso, en el escritorio). Para terminar la instalación basta con trasladar el fichero .class al directorio de *plugins* de ImageJ (C:\Program Files\ImageJ\plugins) y reiniciar el programa (o hacer *click* en *Refresh Menus*, menú *Help*). El *plugin* aparecerá en la parte inferior del menú *Plugins*:

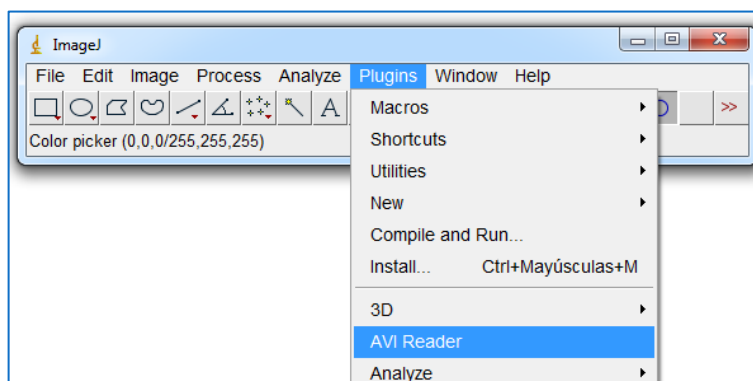
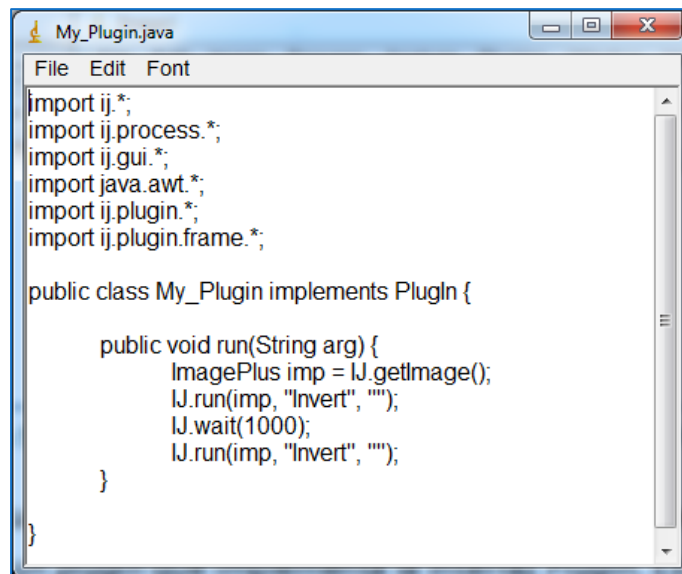


Figura 5.4.13. *Plugin AVI Reader* en la parte inferior del menú *Plugins*

La herramienta bajo estudio, *Plugin*, del submenú *New*, despliega la ventana de edición con el código del prototipo de un *plugin* que implementa la interfaz *PlugIn*. Este tipo de *plugins* sirven para abrir, capturar o generar imágenes. El código java proporcionado tiene el siguiente aspecto (Figura 5.4.14):



```
My_Plugin.java
File Edit Font
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.*;
import ij.plugin.frame.*;

public class My_Plugin implements PlugIn {

    public void run(String arg) {
        ImagePlus imp = IJ.getImage();
        IJ.run(imp, "Invert", "");
        IJ.wait(1000);
        IJ.run(imp, "Invert", "");
    }

}
```

Figura 5.4.14. Código del prototipo de *plugin* que implementa la interfaz *PlugIn*, el cual se presenta en la ventana de edición al hacer *click* en *Plugin*

Las primeras líneas del código están destinadas a importar los paquetes con las clases necesarias para que funcione el código: todas las de *ij* (la biblioteca de ImageJ), todas las de *ij.process*, las de *java.awt*,... La siguiente sentencia marca el comienzo de la clase *My_Plugin* (es necesario que el nombre del *plugin* contenga al menos una barra baja para que ImageJ lo reconozca y lo sitúe en la parte inferior del menú *Plugins*) y además declara que la clase implementa la interfaz *PlugIn*. Consultando la API de ImageJ se puede observar que, efectivamente, esta es la interfaz que tienen que implementar aquellos *plugins* destinados a abrir, capturar o generar imágenes, y que su implementación obliga a programar el método *void run(java.lang.String arg)*, el cual aparece programado en la Figura 5.4.14. Éste método es el que se ejecuta cuando se carga el *plugin*. Este sencillo *plugin* de ejemplo invierte el valor de los píxeles ($v = 255 - u$), espera 1 segundo y los vuelve a invertir. Esto es lo que describen las siguientes sentencias. La primera emplea el método estático de la clase de utilidades *static ImagePlus IJ.getImage()* para guardar la imagen activa en el objeto *ipm* de la clase *ImagePlus*. Esta clase empaqueta un objeto de la clase *ImageProcessor*, que es el que contiene la información de los píxeles de la imagen, así como los metadatos de la imagen y la propia imagen (el objeto *java.awt.Image* que se ve), así que se puede asociar a la imagen en todos los sentidos. La siguiente sentencia utiliza de nuevo un método estático de la clase *IJ*, *static void IJ.run(java.lang.String command, java.lang.String options)*, para ejecutar sobre la imagen directamente un comando de ImageJ, *Invert*, que invierte los píxeles de la imagen como se ha explicado anteriormente. Con la siguiente sentencia se utiliza el método *static void IJ.wait(int msec)* para introducir un retardo de un segundo, y para terminar se vuelve a invertir la imagen.

5.4.5. Plugin Filter

La herramienta *Plugin Filter* abre la ventana de edición con el código del prototipo de un *plugin* de tipo filtro. Estos *plugins*, que implementan la interfaz *PlugInFilter*, están dedicados a procesar la imagen activa.

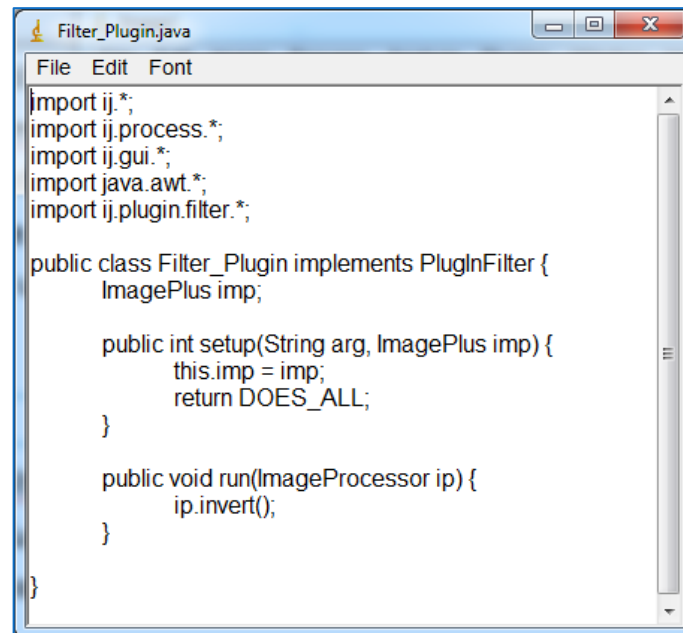


Figura 5.4.15. Código del prototipo de *plugin* que implementa la interfaz *PlugInFilter*, el cual se presenta en la ventana de edición al hacer *click* en *Plugin Filter*

Las primeras líneas de nuevo se utilizan para importar los paquetes con las clases necesarias. La línea que las sigue sirve para abrir la clase, *Filter_Plugin*, y para implementar la interfaz *PlugInFilter* que, como ya se ha comentado, es la que deben implementar los *plugins* orientados al procesamiento de imagen. Para implementar esta interfaz hay que programar dos métodos: *int setup(java.lang.String arg, ImagePlus imp)* y *void run(ImageProcessor ip)*. El primero de ellos se llama una vez al cargar el *plugin* y sirve para configurarlo. El primero de sus argumentos es una cadena de caracteres y el segundo es un objeto de la clase *ImagePlus*, que ya se ha comentado, y que corresponde a la imagen activa. Este método devuelve un *flag* de tipo *int* que especifica las capacidades del *plugin* (e.g., el tipo de imagen que puede procesar). El segundo de los métodos es el que contiene el código correspondiente al procesamiento. Como parámetro se le pasa el objeto de la clase *ImageProcessor* (que, como ya se ha comentado, contiene la información de los píxeles de una imagen) correspondiente al objeto *ImagePlus* de la imagen activa.

La siguiente línea sirve para declarar el objeto de la clase *ImagePlus* que se asociará a la imagen activa: *imp*. Le sigue el código del método *setup* del *plugin*. En él, sencillamente se asigna al objeto *imp* el objeto *ImagePlus* que se pasa por parámetro (es decir, la imagen activa). Para terminar, se devuelve el *flag*, *DOES_ALL*, lo cual indica que el *plugin* es capaz de procesar cualquier tipo de imagen. A continuación aparece el código del segundo método, el procesamiento, que también es muy sencillo. Simplemente se invierten los píxeles de la

imagen activa, aunque a diferencia del *plugin* del apartado anterior, en este caso no se emplean métodos para ejecutar un comando de ImageJ (*Invert*), sino que directamente se emplea el método *invert()* de los objetos de la clase *ImageProcesor*.

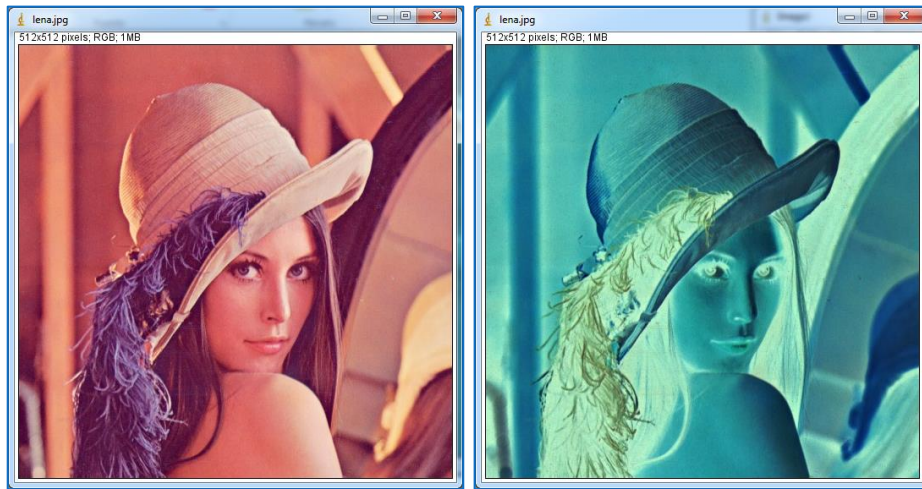


Figura 5.4.16. Izquierda: imagen original (lena.jpg). Derecha: imagen original tras aplicar el *plugin* anterior

5.4.6. Plugin Frame

La herramienta *Plugin Frame* abre la ventana de edición con el código del prototipo de un *plugin* de tipo *frame* (marco o ventana), tal y como se muestra en Figura 5.4.17. Estos *plugins*, que extienden la clase *PlugInFrame*, generan ventanas con barras, botones y otros controles.

Como siempre, las primeras líneas sirven para importar los paquetes con las clases necesarias. A continuación se abre la clase, *Plugin_Frame*, que, como se indica a continuación, extiende la clase *PlugInFrame*. En las siguientes líneas se sobrescribe el constructor de la superclase. El nuevo constructor consta, en primer lugar, del constructor original, al que se le pasa el nombre de la ventana que se abrirá al cargar el *plugin*: *super("Plugin_Frame")*. A continuación se declara e inicializa un área de texto de 15 filas y 50 columnas (*ta*, objeto de la clase *TextArea*, paquete *java.awt*), y se añade a la nueva ventana con el método *add()*, que los objetos de la clase *PlugInFrame* (y subclases, como en este caso) heredan de la clase *Container* (del paquete *java.awt*). Después se empaqueta todo con el método *pack()*, que se hereda de la clase *Window* (de nuevo del paquete *java.awt*). Para terminar, se hace uso de las utilidades de la clase GUI del paquete de ImageJ (*ij.gui.GUI*): se utiliza el método estático *static void center(java.awt.Window w)* para centrar la ventana. Se termina mostrando la ventana con *show()* (ver Figura 5.4.18).

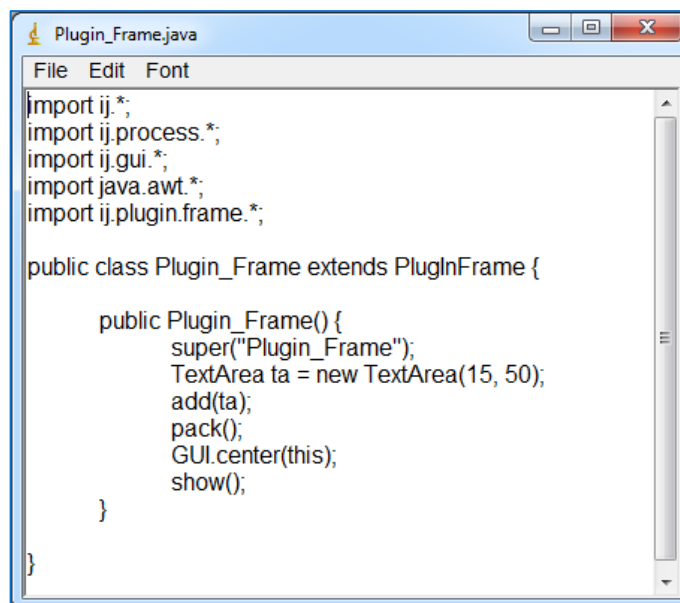


Figura 5.4.17. Código del prototipo de *plugin* que extiende la clase *PlugInFrame*, el cual se presenta en la ventana de edición al hacer *click* en *Plugin Frame*

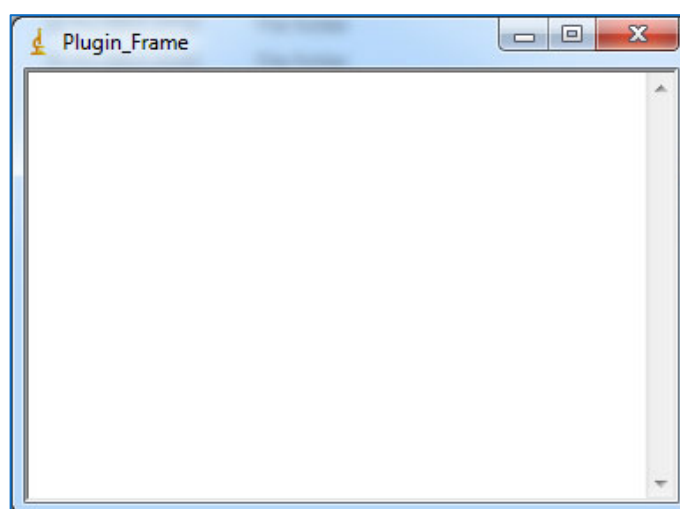
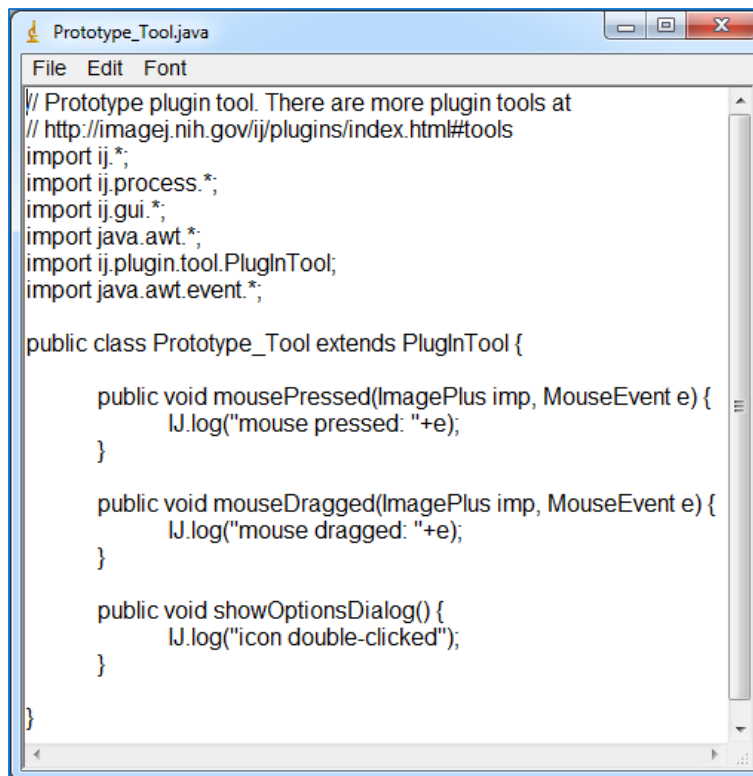


Figura 5.4.18. Resultado de ejecutar el *plugin* anterior

5.4.7. Plugin Tool

La herramienta *Plugin Tool* abre la ventana de edición con el código del prototipo de un *plugin* de tipo herramienta (ver Figura 5.4.19), que son aquellos que se integran en la barra de herramientas de ImageJ (aparecen representados por un icono). El *plugin* cuyo código aparece al pulsar *Plugin Tool* sirve para registrar en la ventana *Log* varios tipos de eventos que el cursor puede llevar a cabo sobre la imagen activa. Su código comienza, como siempre, con las sentencias de importación de paquetes con las clases necesarias. Después se abre la clase, *Prototype_Tool*, y se indica que extiende la clase abstracta *PlugInTool*. A continuación se sobrescriben algunos de sus métodos relacionados con eventos del cursor sobre la imagen activa y se utilizan métodos estáticos de la clase *IJ* para registrarlos en la ventana *Log*.



```
// Prototype plugin tool. There are more plugin tools at
// http://imagej.nih.gov/ij/plugins/index.html#tools
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.tool.PluginTool;
import java.awt.event.*;

public class Prototype_Tool extends PluginTool {

    public void mousePressed(ImagePlus imp, MouseEvent e) {
        IJ.log("mouse pressed: "+e);
    }

    public void mouseDragged(ImagePlus imp, MouseEvent e) {
        IJ.log("mouse dragged: "+e);
    }

    public void showOptionsDialog() {
        IJ.log("icon double-clicked");
    }

}
```

Figura 5.4.19. Código del prototipo de *plugin* tipo herramienta que se presenta en la ventana de edición al hacer *click* en *Plugin Tool*

El resultado de ejecutar este *plugin* se muestra a continuación, en la Figura 5.4.20:

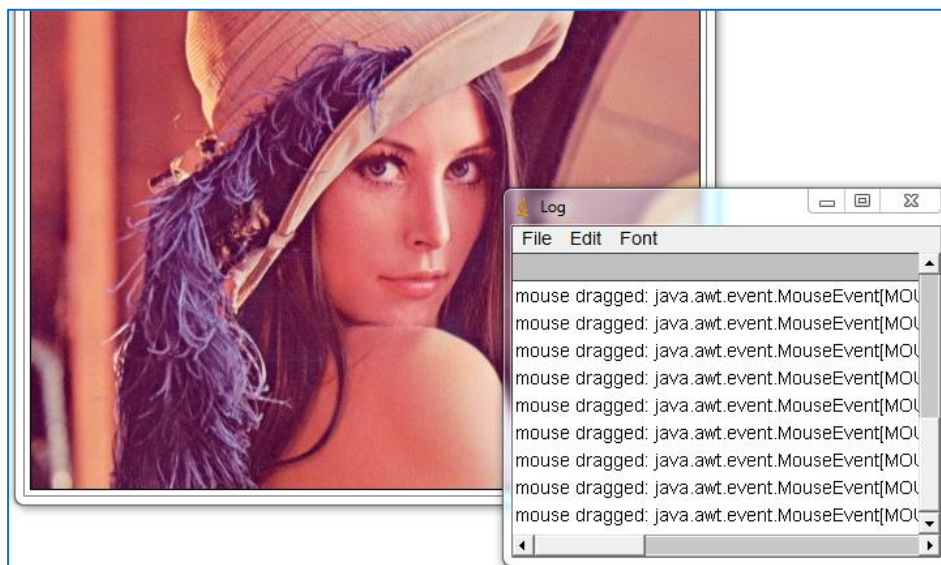


Figura 5.4.20. Imagen junta a la ventana *Log* con el registro de varios eventos de ratón, resultado de ejecutar el *plugin* anterior

5.4.8. Text Window

Esta herramienta permite generar una ventana de texto en blanco. A través de un pequeño menú se puede especificar su nombre, su ancho (en caracteres) y su alto (en líneas), y además se puede determinar si se le

añade una barra de herramientas (marcando la casilla *Menu Bar*) y si los caracteres usados son monoespaciados o no (marcando la casilla *Monospaced Font*).

5.4.9. Table

La herramienta *Table* genera una tabla en blanco. Con un menú emergente se pueden especificar su nombre y sus dimensiones.

5.5. Compile and Run

Esta herramienta permite compilar y ejecutar archivos .java y .class respectivamente.

5.6. Install

Esta última herramienta del menú *Plugins* proporciona una forma alternativa de instalar *macros*, *scripts* o *plugins* en cualquiera de sus formatos. Tan sólo hay que elegir el fichero correspondiente a través de un menú emergente e ImageJ se ocupará de trasladarlo a los directorios de *macros*, *scripts* y *plugins* de ImageJ (y además se hace cargo de las conversiones de formato necesarias y hasta de las compilaciones si se proporcionan ficheros .java, todo de manera transparente al usuario).

6. CONCLUSIONES

En este documento se ha tratado de revisar todas las herramientas de tratamiento digital de imagen que ofrece ImageJ (versión 1.48q), analizando su funcionamiento y prestando especial atención a todos los mecanismos que subyacen a éste. Se han estudiado cuatro grandes grupos de herramientas, que coinciden con los menús que presenta la interfaz de usuario de ImageJ: herramientas de manipulación de imágenes en general (menú *Image*), herramientas de procesado (menú *Process*), herramientas de análisis (menú *Analyze*) y herramientas de extensibilidad (menú *Plugins*).

Una vez examinadas todas estas herramientas se puede concluir que, si bien ImageJ está relativamente orientado al tratamiento digital de imagen en el ámbito de las ciencias de la salud, es lo suficientemente amplio y potente para ser utilizado en el tratamiento de la imagen en general, en cualquier entorno, aunque especialmente en aquellos de carácter científico o técnico, lo cual lo convierte en un *software* del que se podría sacar partido en la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación de la Universidad Politécnica de Madrid. Esto se concluye teniendo en cuenta únicamente el núcleo de ImageJ, pero es aún más cierto si cabe si se considera la que probablemente sea la mayor de sus virtudes: su prácticamente ilimitada extensibilidad. Los *macros*, los *scripts* y especialmente los *plugins* programados en lenguaje Java a partir de la API que ofrece, hacen que ImageJ pueda crecer hasta llegar a solucionar casi cualquier problema de tratamiento digital de imagen. Más aún en un entorno como la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, en el que se combinan el conocimiento del tratamiento digital de la imagen y de la programación en lenguaje Java, así como otras muchas áreas de las telecomunicaciones relacionadas de forma más indirecta.

Además del alcance potencial que tiene ImageJ gracias a su extensibilidad, y de todas las herramientas que incluye por defecto y que son inmediatamente aplicables en materias de tratamiento digital de imagen en la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, ImageJ introduce conceptos novedosos (o al menos no tan comunes en la escuela) como, por ejemplo, el manejo de imágenes mediante pilas e hiperpilas, las imágenes compuestas o las calibraciones espaciales y de densidad. También incluye algoritmos concretos y utilidades a tener en cuenta, como las autoumbralizaciones, los recubrimientos, las operaciones avanzadas con imágenes binarias, el análisis automático de partículas en imágenes binarias, las utilidades para el análisis de geles electroforéticos, y un largo etcétera. Y antes de terminar hay que mencionar otras ventajas que no tienen carácter técnico pero que no por ello tienen menos importancia: ImageJ es un *software* libre, gratuito y de código abierto, y además cuenta con una amplia comunidad de desarrolladores y seguidores que consiguen que mejore y crezca a diario. Así que, en definitiva, ImageJ es, al menos en mi opinión, un programa recomendable que puede resultar beneficioso para la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación de la Universidad Politécnica de Madrid.

7. REFERENCIAS

- [1] «The Median Cut Algorithm,» [En línea]. Available: <http://www.mi.uni-koeln.de/c/mirror/www.cs.curtin.edu.au/units/cg351-551/notes/lect2r1.html>. [Último acceso: 2013].
- [2] «Contrast Stretching,» 2003. [En línea]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm>. [Último acceso: 22 Febrero 2014].
- [3] Fiji Is Just ImageJ, [En línea]. Available: http://fiji.sc/wiki/index.php/Auto_Threshold. [Último acceso: 2013].
- [4] Wikipedia, «Espacio de color Lab,» Fundación Wikimedia, 13 Marzo 2013. [En línea]. Available: http://es.wikipedia.org/wiki/Espacio_de_color_Lab. [Último acceso: 21 Febrero 2014].
- [5] A. Martín Marcos y M. M. Pilar, Colorimetría, Madrid: Editorial Ciencia 3, 1996.
- [6] W. Rasband y T. Ferreira, «ImageJ User Guide,» 2 Octubre 2012. [En línea]. Available: <http://rsbweb.nih.gov/ij/docs/guide/index.html>. [Último acceso: 2013].
- [7] ImageJ Information and Documentation Portal, «ImageJ Information and Documentation Portal,» [En línea]. Available: <http://imagejdocu.tudor.lu/doku.php?id=gui:image:stacks>. [Último acceso: Noviembre 2013].
- [8] J. L. Díaz López, Tratamiento digital de la imagen, vol. II, E. d. Telecomunicación, Ed., Madrid: Universidad Politécnica de Madrid, 1996.
- [9] Wikipedia, «JavaScript,» Fundación Wikimedia, 10 Febrero 2014. [En línea]. Available: <http://es.wikipedia.org/wiki/JavaScript>. [Último acceso: 10 Febrero 2014].
- [10] Wikipedia, «Complemento (informática),» Fundación Wikimedia, 30 Enero 2014. [En línea]. Available: [http://es.wikipedia.org/wiki/Complemento_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Complemento_(inform%C3%A1tica)). [Último acceso: 7 Febrero 2014].